



Nelson Jorge Rocha
Marques Cardoso

**Aproveitamento do Exercício Físico como Fonte de
Energia Renovável**



**Nelson Jorge Rocha
Marques Cardoso**

**Aproveitamento do Exercício Físico como Fonte de
Energia Renovável**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Dr. António Ferreira Pereira de Melo, Professor Catedrático do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Dr. Rui Manuel Escadas Ramos Martins, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

o júri

presidente

Prof. Dr. Alexandre Manuel Moutela Nunes da Mota
Professor Associado da Universidade de Aveiro

arguente principal

Prof. Dr. Carlos Alberto Caridade Monteiro e Couto
Professor Catedrático do Departamento de Electrónica Industrial da Universidade do Minho

orientador

Prof. Dr. António Ferreira Pereira de Melo
Professor Catedrático da Universidade de Aveiro

co-orientador

Prof. Dr. Rui Manuel Escadas Ramos Martins
Professor Auxiliar da Universidade de Aveiro

agradecimentos

Gostava de agradecer ao meu Orientador, Professor Doutor António Ferreira Pereira de Melo, pelo constante apoio e motivação que me deu ao longo de todo este projecto, estando sempre disponível para ajudar e para me guiar neste trabalho. Agradeço também ao meu Co-Orientador, Professor Doutor Rui Manuel Escadas Ramos Martins e a todas as pessoas do Departamento de Electrónica, Telecomunicações e Informática que ao longo destes últimos contribuíram não só para a minha formação a nível profissional mas também pessoal.

Gostava de expressar um agradecimento especial ao Sr. António Neto, cuja experiência e criatividade permitiram a construção deste protótipo.

Finalmente um agradecimento aos meus pais, irmãos, namorada e amigos pela força e motivação que me foi dada ao longo deste trabalho.

palavras-chave

Aparelhos de exercício, energia renovável, bicicleta, gerador DC, inversor, topologia push-pull, microcontrolador, controlo.

resumo

Este trabalho pretende ser um contributo para a área das Energias Renováveis. A utilização dos equipamentos de exercício físico que se encontram em ginásios comuns gera uma determinada quantidade de energia que é desperdiçada. A ideia chave deste projecto é aproveitar parte dessa energia e inseri-la na rede eléctrica. Para esse efeito construiu-se um protótipo baseado num rolo de treino que acciona um gerador DC onde se pode acoplar qualquer bicicleta comum e produzir energia eléctrica. Numa segunda fase procedeu-se à simulação, dimensionamento e construção de um inversor *true sinewave*, com saída de 230 VAC, baseado na topologia *Push-Pull*. O inversor deve receber a tensão DC gerada na bicicleta e transformá-la numa tensão sinusoidal síncrona pronta para ser inserida na rede pública. O interface e controlo do inversor baseiam-se num microcontrolador de 16 bits e 30 MIPS. Foram realizados vários testes em malha aberta, com diferentes cargas, de modo a melhorar a eficiência e estabilidade do inversor.

keywords

Exercise equipment, renewable energy, bicycle, DC generator, inverter, push-pull topology, microcontroller, control.

abstract

The present work intends to be a contribution to improve the Renewable Energies technologies. The use of fitness equipment, which is commonly founded in gyms, generates a certain amount of energy that is wasted. The main idea of this project is to take advantage of this energy and insert it into the AC mains. Therefore a prototype based on a training roller with a DC generator was built, which can engage any ordinary bicycle and produce electricity. A second step involved the simulation, design and construction of a 230 VAC true sinewave inverter, based on the Push-Pull topology. The inverter must receive the DC voltage generated on the bike and turn it into a synchronous sinusoidal voltage ready to be inserted in the AC mains. The interface and control of the inverter are based on a 16-bit microcontroller with 30 MIPS. Several tests have been made in an open-loop configuration with several loads in order to improve the efficiency and stability of the inverter.

Índice

Lista de Figuras	2
Lista de Tabelas	5
Lista de Acrónimos	5
1. Introdução	7
1.1. Motivação	7
1.2. Exercício Físico	7
1.3. Energias Renováveis	10
1.4. Objectivos e Organização da Tese.....	12
2. Estado da Arte	13
2.1. Aparelhos de Exercício Físico	13
2.2. Inversores Comerciais	16
3. Elementos Eléctricos/Electrónicos	21
3.1. Geradores DC	21
3.1.1. Máquinas DC: conceitos básicos.....	21
3.1.2. Princípio de funcionamento.....	21
3.1.3. Factores de perdas dos geradores DC.....	24
3.1.4. Excitação do campo magnético	25
3.1.5. Caracterização dos geradores DC.....	27
3.2. Transformadores	27
3.2.1. O transformador ideal.....	27
3.2.2. Modelo equivalente do transformador.....	29
3.2.3. O uso de materiais ferromagnéticos em transformadores	30
3.2.4. Perdas no núcleo	34
3.2.5. O efeito dos materiais do núcleo na largura de banda.....	35
3.3. Topologias de Inversores e Suas Características.....	35
3.3.1. Topologias de inversores	35
3.3.2. Princípio de funcionamento do inversor <i>Push-Pull</i>	40
3.3.3. Análise da tensão de entrada e saída no transformador.....	43
3.3.4. Técnicas de modulação para inversores	46
3.4. Snubbers	49
4. Modelo de Simulação	53
4.1. O Esquema e Seus Elementos	53
4.2. Resultados da Simulação	58
5. Construção do Protótipo	65
5.1. Rolo de Treino com Gerador DC	65
5.2. Circuito Inversor.....	68
6. Apresentação de Resultados	79
6.1. O Software.....	79
6.2. Teste de Funcionamento.....	80
6.3. Desempenho	86
7. Conclusões e Trabalhos Futuros	91
Referências	93

Lista de Figuras

Figura 1. 1 - Exemplo de um indivíduo a pedalar	9
Figura 1. 2 - Fontes de energia renovável	10
Figura 1. 3 - Sistema de microgeração doméstico	11
Figura 2. 1 - Máquina elíptica E8300 da SportsArt	14
Figura 2. 2 - Bicicleta de treino C5150 da SportsArt	14
Figura 2. 3 - Climber 7005 da SportsArt	14
Figura 2. 4 - Ergómetro 2100 da SportsArt	15
Figura 2. 5 - Passadeira 6320 da SportsArt	15
Figura 2. 6 - Exemplo de LCD na 8010UX da Stex	16
Figura 2. 7 - Detalhe do minigerador	16
Figura 2. 8 - Inversor <i>stand-alone</i> da Honestar	17
Figura 2. 9 - Sistema <i>stand-alone</i> baseado em painéis solares	17
Figura 2. 10 - Sistema <i>grid-connected</i> baseado em painéis solares	18
Figura 3. 1 - Modelo de um gerador DC básico	22
Figura 3. 2 - Funcionamento de um gerador DC	22
Figura 3. 3 - Detalhe do colector e escovas	23
Figura 3. 4 - Acção do comutador	24
Figura 3. 5 - Esquema excitação série	26
Figura 3. 6 - Curva de carga da excitação série	26
Figura 3. 7 - Excitação paralela	26
Figura 3. 8 - Curva de carga da excitação paralela	26
Figura 3. 9 - Excitação composta	26
Figura 3. 10 - Curva de carga da excitação composta	26
Figura 3. 11 - Princípio do transformador	28
Figura 3. 12 - Modelo equivalente do transformador real	29
Figura 3. 13 - Curva de histerese	31
Figura 3. 14 - Exemplo de alguns núcleos de ferro	32
Figura 3. 15 - Exemplo de alguns núcleos de Ferrite	33
Figura 3. 16 - Regulador linear	36
Figura 3. 17 - Regulador comutado	36
Figura 3. 18 - Conversor <i>Flyback</i> com transformador de isolamento	36
Figura 3. 19 - Principais formas de onda do conversor <i>Flyback</i>	37
Figura 3. 20 - Inversor <i>Half-Bridge</i>	38
Figura 3. 21 - Inversor <i>Full-Bridge</i>	39
Figura 3. 22 - Inversor <i>Push-Pull</i>	40
Figura 3. 23 a) Funcionamento do <i>Push-Pull</i>	42
Figura 3. 24 - Formas de onda do <i>Push-Pull</i>	43
Figura 3. 25 - Forma de onda da tensão de entrada	44
Figura 3. 26 - Forma de onda da tensão de saída	46
Figura 3. 27 - Onda quadrada e <i>modified sinewave</i>	47
Figura 3. 28 - Modulação por SPWM	47
Figura 3. 29 - SPWM com maior frequência de comutação	48

Figura 3. 30 - Espectro do SPWM para $n=15$ e $m=0,8$	48
Figura 3. 31 - Forma de onda aos terminais de Q1	49
Figura 3. 32 - Snubber RC	50
Figura 3. 33 - Snubber RCD	50
Figura 3. 34 - Q1 com e sem snubber	51
Figura 4. 1 – Diagrama de blocos do modelo de simulação	54
Figura 4. 2 - Circuito Pspice do modelo de simulação	55
Figura 4. 3 - Detalhe do gerador de PWM	56
Figura 4. 4 - Detalhe do transformador	56
Figura 4. 5 - Forma de onda da rede usada neste modelo	57
Figura 4. 6 - Comportamento dinâmico do modelo de simulação	59
Figura 4. 7 - Pormenor da modulação por SPWM	60
Figura 4. 8 - Formas de onda de outros pontos do circuito	61
Figura 4. 9 - Resultado da simulação com perturbação da tensão de entrada	63
Figura 5. 1 – Protótipo do rolo de treino já com uma bicicleta montada	65
Figura 5. 2 - Detalhe do gerador DC e demais elementos	66
Figura 5. 3 - Detalhe dos braços de apoio e veios de fixação	66
Figura 5. 4 - Pormenor de uma das quatro bases de nivelamento	67
Figura 5. 5 - Roda de inércia do rolo de treino	67
Figura 5. 6 - Esquema do microcontrolador	69
Figura 5. 7 - Esquema da fonte de alimentação de 5V	70
Figura 5. 8 - Esquema do inversor	71
Figura 5. 9 - Esquema do detector de passagem por zero	72
Figura 5. 10 - Placa de circuito impresso	76
Figura 5. 11 - Aspecto final do circuito inversor	77
Figura 6. 1 - Detalhe da aplicação de controlo feita em Visual Studio 2008	80
Figura 6. 2 - Onda de corrente (CH1) e tensão (CH2) para $V_{dc}=15V$, $f=20KHz$, carga=25W	81
Figura 6. 3 - Onda de corrente (CH1) e tensão (CH2) para $V_{dc}=15V$, $f=20KHz$, carga=60W	81
Figura 6. 4 - Onda de corrente (CH1) e tensão (CH2) para $V_{dc}=20V$, $f=20KHz$, carga= 60W	81
Figura 6. 5 - Sinal de SPWM em Q1 (CH1) e Q2 (CH2) para $V_{dc}=15V$, $f=20KHz$, carga=25W	83
Figura 6. 6 - Detalhe do sinal de SPWM em Q1 (CH2) para $V_{dc}=15V$, $f=20KHz$, carga=25W	83
Figura 6. 7 - Tensão de dreno (CH1) e gate (CH2) de Q1 (sem snubber)	84
Figura 6. 8- Tensão de dreno (CH1) e gate (CH2) de Q1 (com snubber)	84
Figura 6. 9 - Corrente de dreno (CH1) e tensão de gate (CH2) em Q1	84
Figura 6. 10 - Tensão de saída (CH2) e detector de passagem por zero (CH1)	85
Figura 6. 11 - Onda de corrente (CH1) e tensão (CH2) para V_{dc} =gerador, $f=20KHz$, carga=25W	86
Figura 6. 12 - Onda de corrente (CH1) e tensão (CH2) para V_{dc} =gerador, $f=20KHz$, carga=60W	86

Figura 6. 13 - Onda de corrente (CH1) e tensão (CH2) para V_{dc} =gerador, $f=20\text{Khz}$, carga=100W	87
---	----

Lista de Tabelas

Tabela 4. 1 – Eficiência em função da tensão do gerador	62
Tabela 6. 1 - Eficiência em função da carga de saída.....	88
Tabela 6. 2 - Eficiência em função da tensão de entrada com lâmpada de 60W	88
Tabela 6. 3 - Eficiência em função da frequência de comutação com lâmpada de 60W	88

Lista de Acrónimos

AC	- Alternating Current
DC	- Direct Current
LCD	- Liquid Crystal Display
MIPS	- Milhões de Instruções por Segundo
MOSFET	- Metal Oxide Semiconductor Field Effect Transistor
MPPT	- Maximum Power Point Tracker
PI	- Proporcional Integrador
PID	- Proporcional Integrador e Derivativo
PLL	- Phase-Locked Loop
PWM	- Pulse-Width Modulation
SPWM	- Sinusoidal Pulse-Width Modulation
UPS	- Uninterruptible Power Supply

1. Introdução

1.1. Motivação

A importância da energia no mundo moderno é inquestionável. Desempenha um papel fundamental na indústria, transportes, serviços e no dia-a-dia das pessoas. O seu consumo tem aumentado de ano para ano, fruto do crescimento demográfico e do rápido desenvolvimento das economias emergentes, como é o caso da China e da Índia. No entanto, o modelo energético actual, baseado sobretudo nos combustíveis fósseis, não é económica e ambientalmente sustentável a longo prazo. É precisamente por essa razão que as energias renováveis têm vindo a ganhar relevância e demonstra ser uma solução alternativa e sustentável. Neste momento é uma área em plena expansão e domina as agendas políticas dos países mais desenvolvidos.

O exercício físico, por sua vez, ocupa uma posição de destaque nas sociedades modernas. O sedentarismo aliado a uma má alimentação, rica em gorduras saturadas e açúcares refinados, reflecte o estilo de vida actual das pessoas, especialmente nos centros urbanos e áreas fortemente industrializadas. A actividade física ajuda a reduzir esses efeitos e contribui de forma decisiva para a saúde física e mental dos seus praticantes. A crescente procura por actividade física levou a que nos últimos anos surgissem inúmeros ginásios e, com eles, um vasto leque de equipamentos para exercício. A grande maioria desses equipamentos não aproveita nenhuma da energia que se produz quando são utilizados.

Este trabalho estabelece uma ponte que liga estas duas áreas ao criar um novo conceito de utilização da energia gerada num aparelho de exercício, de modo a que possa ser aproveitada. Este conceito não se limita apenas aos equipamentos de exercício e pode facilmente estender-se a outros domínios e aplicações. Constitui, portanto, um desafio aliciante e uma oportunidade para se poder aplicar as técnicas e ferramentas que a Engenharia disponibiliza na resolução de problemas e na criação de soluções.

1.2. Exercício Físico

O exercício físico pode ser definido como um conjunto de actividades motoras realizadas pelo corpo humano com o intuito de melhorar ou manter a aptidão física geral, contribuindo de forma importante para a saúde e bem-estar das pessoas. O exercício físico pode ter uma vertente mais séria ou profissional, no caso dos atletas de alta competição mas, de uma maneira geral, é tido como fonte de lazer e recreação. Promove

a regeneração e desenvolvimento dos tecidos musculares, tem um impacto positivo no sistema cardiovascular e respiratório e estimula o sistema imunológico. Também ajuda a prevenir doenças cardíacas e diabetes, moderando a concentração de colesterol no sangue ao mesmo tempo que reduz a obesidade. A somar a estes benefícios está também o fortalecimento da saúde mental.

Com todos estas vantagens, não é de admirar que governos de todo o mundo se empenhem em divulgar e implementar programas que promovam a actividade física junto das populações, com o objectivo de combater o sedentarismo. Considerada a doença do século, o sedentarismo reflecte o estilo de vida moderno e tem repercussões negativas a longo prazo, quer a nível de saúde pública quer a nível da economia.

Esta generalização do conceito que o exercício físico promove o bem-estar e a saúde levou a que os últimos anos se multiplicassem a instalação de inúmeros ginásios (também conhecidos por *Health Club*), especialmente junto dos centros urbanos. Acompanhando a crescente procura da actividade física por parte da população, estes autênticos “centros comerciais” de exercício físico têm-se tornado cada vez maiores e mais completos no que toca à oferta de serviços.

O panorama em Portugal é semelhante, contudo tem-se verificado um crescimento mais ténue quando comparado com outros países europeus. Ainda há um longo caminho a percorrer nesta matéria até atingir níveis como, por exemplo na Finlândia, onde cerca de 70 % da população admite praticar exercício físico e frequentar ginásios com regularidade. Existe, portanto, um óbvio potencial de crescimento desta área no mercado nacional.

Os *Healt Club* modernos caracterizam-se por disponibilizar cada vez mais aparelhos de exercício que, por sua vez, se vão tornando mais sofisticados. Actualmente existe uma grande diversidade de equipamentos desta natureza que proporcionam aos praticantes um conjunto de inúmeras actividades diferentes. Uma parte significativa destas máquinas possui, de alguma forma, sistemas de pesos ou mecanismos de atrito que absorvem a energia mecânica produzida por um indivíduo, durante um determinado exercício. Este facto reveste-se de especial importância no âmbito deste trabalho, na medida que se pode aproveitar parte dessa energia que, de outra forma, é desperdiçada.

Mas a final que quantidade de energia que está aqui em jogo?

Segue-se um pequeno exercício de cálculo que ajuda a responder melhor a esta questão:

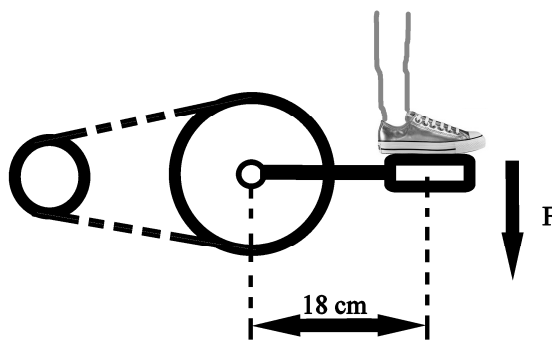


Figura 1. 1 - Exemplo de um indivíduo a pedalar

Considera-se um indivíduo com uma massa corporal de 75 kg a pedalar numa bicicleta, conforme ilustra a Fig 1.1.

Sabendo que:

$$F = m \cdot a \quad (1.1)$$

$$P = F \cdot V \quad (1.2)$$

$$V = \omega \cdot r \quad (1.3)$$

Então obtém-se:

$$P = m \cdot a \cdot \omega \cdot r \quad (1.4)$$

Onde:

P - Potência

m - Massa

a - Aceleração

ω - Velocidade angular

r - Raio

Se o indivíduo pedalar usando uma força equivalente a 60% do seu peso, com uma velocidade angular na pedaleira de π rad/s, verifica-se que a potência dispendida é de:

$$0.6 \times 75 \times 9.8 \times 3.14 \times 0.18 = 249.4 \text{ W com base na Eq. 1.4}$$

Mantendo este ritmo, ao fim de 30 de minutos de exercício obtém-se uma energia acumulada de 448,2 KJ (ou 0,12 KWh). Se se considerarem 30 pessoas a pedalar simultaneamente a um ritmo regular numa ala de um ginásio durante uma hora, a energia produzida começa a ser significativa.

O exemplo anterior mostra claramente as potencialidades de aproveitar parte da energia gerada pelo exercício físico num ginásio comum. Se, por exemplo, alguns dos aparelhos forem alterados de modo a substituir os sistemas de atrito por pequenos

geradores eléctricos e se a energia eléctrica produzida for inserida na rede pública consegue-se, de algum modo, aumentar a rentabilidade económica de um ginásio, cortando nos seus custos fixos. Nesta perspectiva, pode-se considerar o exercício físico como uma fonte de energia renovável.

1.3. Energias Renováveis

O mundo global tal como o conhecemos hoje, depende totalmente das várias formas de energia que o homem dispõe. Praticamente todas as actividades humanas envolvem, de alguma forma, o consumo de energia.

A principal fonte de energia actual provém dos combustíveis fósseis. Esta situação aliada ao grande consumo energético coloca grandes problemas de sustentabilidade ambiental e económica. Ainda recentemente a especulação dos mercados fez disparar o preço do barril de crude e soou o alerta de que se torna imperativo encontrar outras soluções energéticas mais limpas e sustentáveis.

É precisamente em este contexto que se insere o conceito de energia renovável. Energia renovável é aquela que tem origem em fontes naturais e que possuem a capacidade de regeneração, isto é, são virtualmente inesgotáveis. Caracteriza-se também por ter um impacto ambiental menor. São estas qualidades que conferem às energias renováveis uma importante alternativa em relação às fontes fósseis.

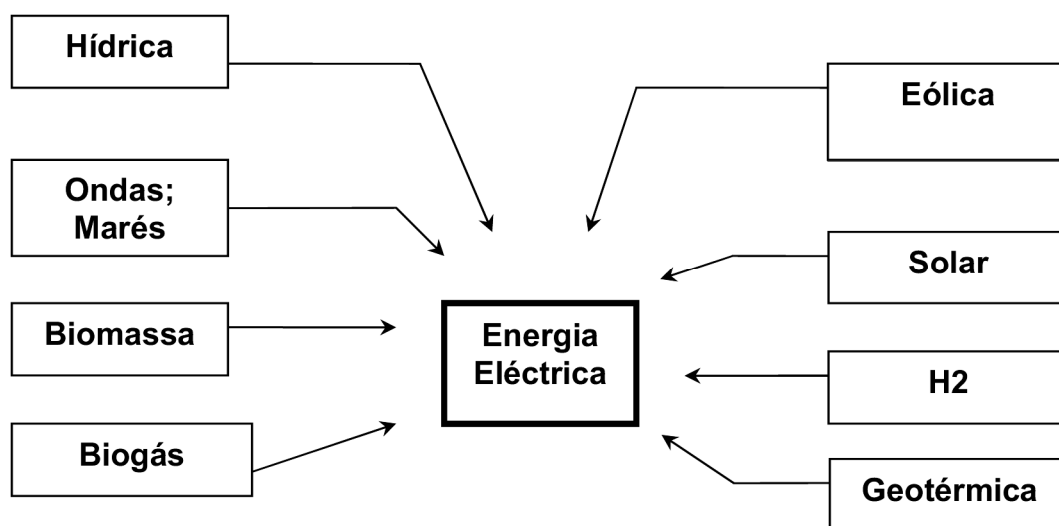


Figura 1. 2 - Fontes de energia renovável

A Fig 1.2 enumera algumas das principais fontes de energia renovável actualmente existentes e que podem servir, por exemplo, para gerar electricidade.

A electricidade é uma fonte de energia limpa, por excelência. Actualmente a energia eléctrica baseia-se em modelos de produção e distribuição centralizados. Só mais recentemente é que vários países (incluindo Portugal) têm apostado na produção

descentralizada como forma de reduzir perdas nas redes de distribuição, de diminuir a dependência externa e de dar ao consumidor um papel mais activo nesta matéria. É com base nesta ideia que nasce o conceito de microgeração. Entendida como sendo a produção de energia eléctrica pelo próprio consumidor, utilizando equipamentos de pequena escala e com a possibilidade dos excedentes energéticos poderem ser vendidos à rede de distribuição pública, afigura-se como a grande tendência do futuro. Em Portugal já foram dados passos importantes no sentido de promover a microgeração baseada em energias renováveis, nomeadamente com a criação de programas de incentivo assim como a implementação de legislação que regula o sector. No entanto, a complexidade burocrática ainda impõe obstáculos que atrasam a sua generalização.

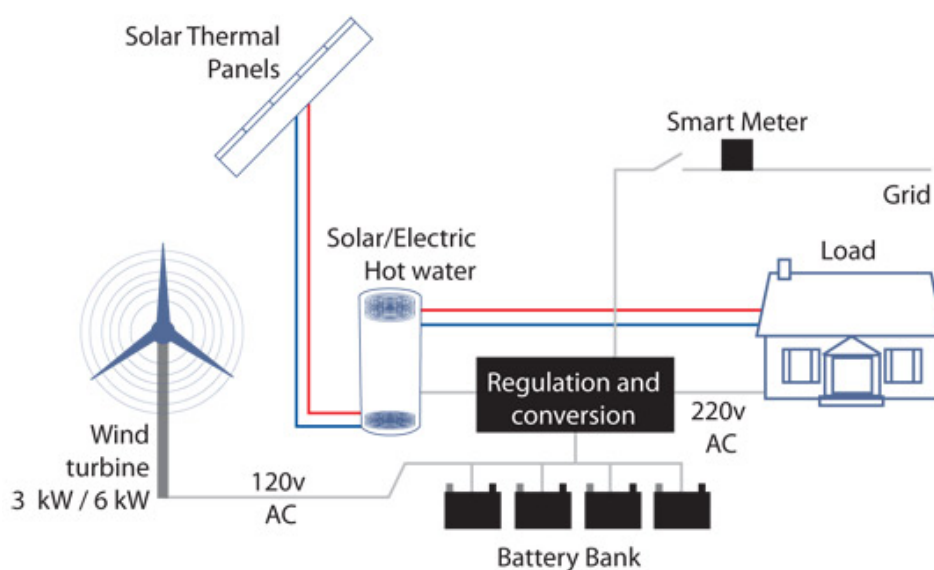


Figura 1.3 - Sistema de microgeração doméstico

A Fig. 1.3 mostra um exemplo da implementação de um sistema de microgeração doméstico. Actualmente existem múltiplas soluções comerciais baseadas sobretudo em painéis solares e pequenas turbinas eólicas. Em locais remotos, onde não chega a distribuição de energia eléctrica, é necessário incluir no sistema de geração um conjunto de baterias para armazenamento energético.

De acordo com as necessidades energéticas ou vantagens económicas, os equipamentos para microgeração descentralizada de electricidade podem ser utilizados de diferentes formas:

- **Consumo base** (*baseload*): O equipamento funciona continuamente à sua potência nominal (rendimento máximo), inferior à potência necessária para satisfazer a totalidade das necessidades.
- **Picos de consumo** (*peak shaving*): O equipamento funciona apenas para cobrir os picos de consumo.

- **Equipamento auxiliar de segurança (*standby*):** Sistema de segurança que permite fornecer a potência total necessária quando ocorrem falhas inesperadas na rede eléctrica.
- **Fornecimento ininterrupto de electricidade (UPS):** Sistema que fornece sempre uma potência constante independentemente do que se passe com a rede eléctrica.
- **Geração portátil:** Sistema de produção de electricidade que facilmente pode ser transportado para diferentes locais. Um exemplo deste tipo de sistema são os painéis solares flexíveis/desdobráveis.

1.4. Objectivos e Organização da Tese

O principal objectivo deste trabalho consiste no estudo, desenvolvimento e teste de um aparelho de exercício físico que inclui um gerador de energia eléctrica e de um inversor *grid-connected* de 100W, para uma tensão de entrada entre 10V e 20V. A energia produzida pelo aparelho será inserida na rede, através do inversor.

O presente documento encontra-se organizado da seguinte forma: o capítulo 2 resume o que existe de mais actual no mercado a nível de aparelhos de exercício físico e inversores; no capítulo 3 apresentam-se as técnicas e os conceitos teóricos mais relevantes para este trabalho; no capítulo 4 procede-se à simulação do inversor e apresentam-se os respectivos resultados; no capítulo 5 descreve-se o processo de construção dos protótipos; no capítulo 6 procede-se ao ensaio do equipamento e apresentam-se os resultados obtidos. As conclusões e trabalhos futuros encontram-se no capítulo 7.

2. Estado da Arte

2.1. Aparelhos de Exercício Físico

Hoje em dia os ginásios modernos disponibilizam aos seus clientes um vasto leque de equipamentos para exercício físico, para além de outros serviços. A diversidade de aparelhos existente advém da necessidade de dividir a actividade física num conjunto muito específico de exercícios, com o objectivo de os focalizar em determinadas partes ou sistemas do corpo humano. Numa perspectiva mais abrangente, as várias máquinas complementam-se entre si.

Apesar das diferenças evidentes, as máquinas de exercício podem ser agrupadas em duas grandes classes: os aparelhos de musculação e os aparelhos cardiovasculares. Como o próprio nome indica, os aparelhos de musculação trabalham a nível dos tecidos musculares e ligamentos enquanto que os aparelhos cardiovasculares incidem os seus exercícios na resistência física, estimulando o sistema cardíaco, circulatório e respiratório. Uma particularidade interessante associada a estas duas classes tem a ver com a natureza mecânica de cada uma delas. No caso das máquinas de musculação, a absorção de energia produzida pelo praticante baseia-se, na maioria dos casos, em mecanismos de contra-pesos e molas. Isto pressupõe a execução de movimentos do tipo “vai-e-vem” dos membros superiores, inferiores ou tronco. Por outro lado, as máquinas cardiovasculares transferem a energia gerada para um sistema de discos rotativos. A energia cinética acumulada nesses discos é depois dissipada através do atrito provocado por cintas de aperto regulável. Essas cintas são montadas em torno do perímetro do disco e a força que exercem pode ser ajustada pelo utilizador. Estas máquinas proporcionam um conjunto de movimentos mais regulares e contínuos ao longo do tempo, o que as torna ideais para serem adaptadas de forma a incluírem um gerador que produza energia eléctrica. A natureza mecânica dos aparelhos de musculação torna aquele tipo de adaptação demasiado complexo e dispendioso para sequer ser considerado. Por essa razão as máquinas cardiovasculares têm um papel central no âmbito deste trabalho.

Os aparelhos cardiovasculares são agrupados com base na funcionalidade e tipo de exercício a que se destinam. Segue-se uma pequena descrição dos principais tipos de máquinas que se podem encontrar no mercado, cujas imagens foram retiradas de [14]:

- **Elíptica:** este aparelho estacionário simula a acção de caminhar ou correr. O facto do peso do corpo estar igualmente distribuído em ambas as pernas do atleta evita uma excessiva pressão a nível dos joelhos. Aqui não há o impacto

do pé no chão associado a uma corrida convencional. Esta máquina possui também um par de apoios para os braços (*handrails*).



Figura 2. 1 - Máquina elíptica E8300 da SportsArt

- **Bicicleta:** este aparelho simula a ação de pedalar. Assemelha-se a uma bicicleta vulgar, contudo não possui rodas e é estática. As suas características mecânicas simples aliadas à natureza do exercício que proporciona, torna este dispositivo ideal para ser adaptado.



Figura 2. 2 - Bicicleta de treino C5150 da SportsArt

- **Climber:** esta máquina estacionária simula a ação de subir degraus. É composta por uma coluna ajustável e um corpo central onde são incorporados um par de pedais basculantes e apoios para os braços.



Figura 2. 3 - Climber 7005 da SportsArt

- **Ergómetro:** este dispositivo estacionário simula a acção de remar. Basicamente o atleta está sentado e puxa repetidamente com os braços, tronco e pernas um volante que provoca tracção num disco de inércia através de um cabo.



Figura 2. 4 - Ergómetro 2100 da SportsArt

- **Passadeira:** este aparelho estacionário simula a acção de caminhar ou correr. Ao contrário da elíptica, na passadeira o atleta tem de colocar alternadamente o peso do corpo em cada uma das pernas, como faz naturalmente. No fundo, a passadeira não é mais do que um tapete rolante accionado por um motor eléctrico de velocidade regulável. O tapete deve oferecer o mínimo de resistência ao acto de correr, sob pena do atleta deslocar-se constantemente para a frente e para fora da máquina. Se se substituísse o motor, que mantém o ritmo da corrida, por um gerador a situação seria ainda pior. Mesmo com mais inclinação, uma passadeira nestas circunstâncias seria extremamente desconfortável e pouco prática. Este é um caso particular das máquinas cardiovasculares em que uma possível adaptação seria inviável.



Figura 2. 5 - Passadeira 6320 da SportsArt

Os aparelhos de exercício mais recentes possuem um vasto número de funcionalidades electrónicas. Vêm equipados com microcontroladores e sensores que permitem medir grandezas como força, velocidade e até o ritmo cardíaco. É possível seleccionar vários programas de treino pré-estabelecidos e obter resultados em tempo real como potência, distância percorrida, consumo de calorias/h, nível de resistência, rpm entre outros. Os melhores modelos disponibilizam display's LCD (Fig. 2.6) e até possuem pequenos geradores (Fig. 2.7) que alimentam esses sistemas electrónicos e não necessitam de serem ligados à rede eléctrica. As cintas de atrito estão a ser substituídas por discos que dissipam a energia através de correntes de Foucault ou resistências.



Figura 2. 6 - Exemplo de LCD na 8010UX da Stex



Figura 2. 7 - Detalhe do minigerador

No anexo A (pág. 95) encontra-se o exemplo de um aparelho cardiovascular topo de gama da Stex e as suas especificações mais importantes.

2.2. Inversores Comerciais

Um inversor é um dispositivo de potência que converte a energia eléctrica de DC para AC. A sua arquitectura base interna inclui transformadores, elementos de comutação e circuitos de controlo. As baterias, painéis solares e aerogeradores estão entre as fontes DC vulgarmente mais utilizadas na alimentação dos inversores. No mercado podem ser encontradas duas categorias que classificam este tipo de dispositivos: os inversores *stand-alone* e os inversores *grid-connected*.

Os inversores *stand-alone* caracterizam-se por funcionarem em modo autónomo, isto é, trabalham totalmente separados da rede eléctrica e alimentam cargas de forma independente.

Os modelos mais baratos (Fig. 2.8) podem ser encontrados praticamente em qualquer loja de produtos electrónicos, cuja gama de potências não vá para além de algumas centenas de watt. A tensão de saída destes modelos costuma ser uma sinusóide

aproximada por uma onda quadrada, sendo esta a forma mais simples de modular uma saída AC. É comum necessitarem de uma tensão de entrada regulada e desligarem quando essa tensão cai abaixo de um determinado valor. Tipicamente a regulação de saída destes modelos é feita através do controlo de tensão. São usados fundamentalmente para alimentar pequenos dispositivos eléctricos de baixa potência a partir de um sistema de armazenamento baseado em baterias.

Os modelos *stand-alone* de maior potência (e portanto mais dispendiosos) já permitem, por exemplo, alimentar totalmente uma habitação (Fig. 2.9). Disponibilizam uma tensão de saída *true-sinewave* e encontram-se em aplicações que vão desde sistemas de microgeração domésticos até sistemas integrados em navios. No anexo B (pág. 98) encontra-se o exemplo de um inversor *stand-alone* da SMA Solar Technology e as suas características mais relevantes.



Figura 2. 8 - Inversor *stand-alone* da Honestar

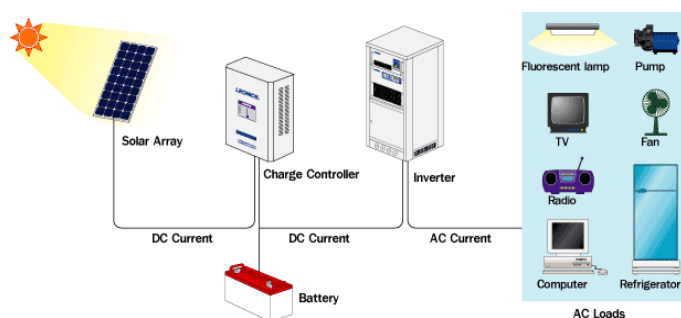


Figura 2. 9 - Sistema *stand-alone* baseado em painéis solares

Os inversores *grid-connected* são caracterizados por estarem ligados permanentemente à rede eléctrica (Fig 2.10). Estes dispositivos devem produzir uma sinusóide de saída com boa qualidade, i.e., com baixa distorção. São projectados de modo a acompanharem a frequência e tensão imposta pela rede. O controlo de potência destes aparelhos é feito com base em circuitos analógicos ou circuitos com microcontroladores. A estratégia de controlo passa por regular a amplitude e fase da forma de onda de saída com o objectivo de controlar o fluxo de potência entre o inversor e a rede. Alguns modelos mais completos possuem um algoritmo de controlo que permite maximizar a transferência de potência entre a fonte DC e a rede (algoritmo *maximum-power-point tracking* ou MPPT). Outra característica interessante destes modelos é a capacidade que têm em absorver ou exportar potência reactiva de modo a regularem o factor de potência de uma determinada instalação, aproximando-o do valor 1. Normalmente esta capacidade de regulação abrange apenas a rede ao nível local (e.g. instalações domésticas).

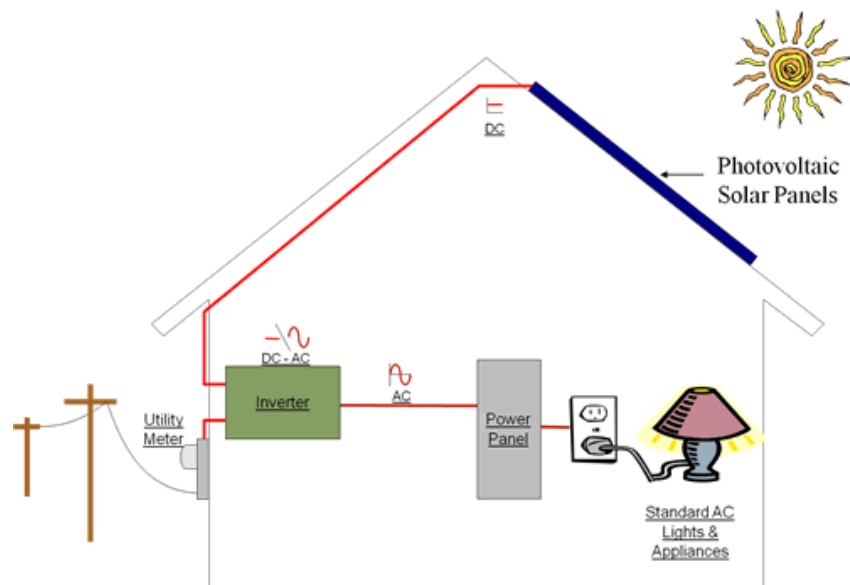


Figura 2. 10 - Sistema *grid-connected* baseado em painéis solares

O tempo de resposta e adaptação dos inversores *grid-connected* actuais é extremamente rápida e depende sobretudo da largura de banda do sistema de controlo. O uso de comutadores baseados na tecnologia moderna de semicondutores permite tempos de resposta na ordem dos milissegundos. Os modelos mais antigos eram baseados em tiristores, mais lentos e com baixa frequência de comutação. Isto levava a que se produzissem harmónicos indesejáveis resultando muitas vezes em perdas, sobreaquecimento e numa potência de saída de pobre qualidade. Os avanços na tecnologia da electrónica de potência permitiu o desenvolvimento de uma nova geração de inversores que produzem uma saída sinusoidal com um nível de harmónicos bastante reduzido e que se baseiam em técnicas de modulação por SPWM.

A sincronização dos inversores actuais é feita de forma automática e tipicamente utilizam um detector de passagem por zero na tensão da rede. Uma vez que este processo de sincronização não envolve qualquer tipo de inércia (ao contrario de um alternador, por exemplo, que pode ser encarado como uma massa rotativa susceptível de ser acelerada ou desacelerada) é possível fazer com que a sinusóide de referência seja corrigida muito rapidamente. Existem outros modelos em que a sincronização é feita através de PLL's (*Phase-Locked Loops*), aumentando a imunidade ao ruído. No entanto, o tempo de resposta deste tipo de sincronização é mais lento quando comparado com o primeiro.

Em caso de avaria, os inversores têm a capacidade de injectar componentes DC indesejáveis na rede. Estas componentes são causadas por algum mau funcionamento dos elementos electrónicos de potência que produzem um meio ciclo positivo que é diferente do meio ciclo negativo. Daí que por vezes se recorra à instalação de um transformador de isolamento entre o inversor e a rede para prevenir essa situação.

Contudo muitos dispositivos mais recentes já possuem sistemas de detecção de componentes DC na sua saída.

Um outro fenómeno relevante a considerar é o *islandig*. *Islanding* é o que acontece quando é desligada (ou falha) a energia de um determinado ramo da rede eléctrica onde estão múltiplos inversores ligados e estes continuam a fornecer potência nesse ramo, mantendo a referência de frequência entre si. Isto pode representar uma condição perigosa se, por exemplo, houver necessidade de fazer algum tipo de intervenção nesse ramo. Apesar de se ter cortado o fornecimento externo, os técnicos podem desconhecer que continua a haver energia fornecida pelos inversores. É possível encontrar no mercado inversores que detectam a condição de *islandig* através da monitorização de vários parâmetros da rede, nomeadamente na medição da sua impedância, variação da frequência (*drift*) ou aumento súbito do número de harmónicos. No anexo C (pág. 100) encontra-se o exemplo de um inversor *grid-connected* assim como as suas características mais importantes.

3. Elementos Eléctricos/Electrónicos

3.1. Geradores DC

3.1.1. Máquinas DC: conceitos básicos

As máquinas de corrente contínua (DC) são essencialmente elementos de conversão de energia eléctrica em mecânica (e vice-versa). Estas máquinas podem funcionar quer como motor, quer como gerador. Motores e geradores DC têm a mesma base de construção, diferindo no sentido da conversão de energia. Para entender melhor o seu funcionamento segue-se uma síntese dos termos básicos mais importantes:

- **Rotor:** é o elemento rotativo da máquina, responsável pela conversão da energia eléctrica/mecânica (caso dos motores) e vice-versa (caso dos geradores). É composto essencialmente por um eixo onde são montadas bobinas de fio de cobre em torno de um núcleo laminado de ferro. Dá-se o nome de induzido ou armadura ao conjunto formado pelas bobinas, núcleo, colector e escovas.
- **Estator:** é o elemento fixo da máquina que gera os pólos do campo magnético. No caso da Fig. 3.1 o campo magnético é gerado por um par de ímanes permanentes. O campo magnético pode ser gerado também por electroímãs, como se mostrará mais adiante. Ao subconjunto de elementos que geram o campo magnético denomina-se indutor ou campo.
- **Colector:** estrutura cilíndrica composta por vários segmentos de material condutor, isolados entre si. Cada um dos enrolamentos que compõem o induzido termina num par desses segmentos.
- **Escova:** é o elemento que promove o contacto eléctrico entre o colector e um circuito externo.

3.1.2. Princípio de funcionamento

São necessárias 3 condições para induzir uma corrente num condutor: um campo magnético, um condutor e um movimento relativo entre os 2. Um gerador DC básico é composto por 4 partes: um campo magnético, um condutor em loop, um comutador e um

par de escovas (Fig. 3.1). O campo magnético pode ser gerado por um ímã permanente ou por um electroímã, como já foi referido. Neste exemplo será usado um ímã permanente para descrever um gerador DC básico.

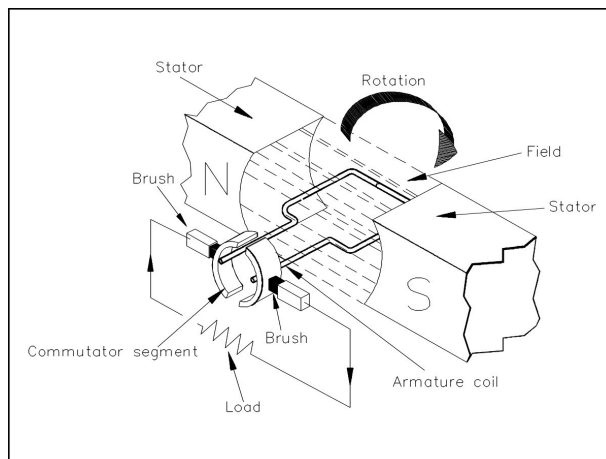


Figura 3. 1 - Modelo de um gerador DC básico

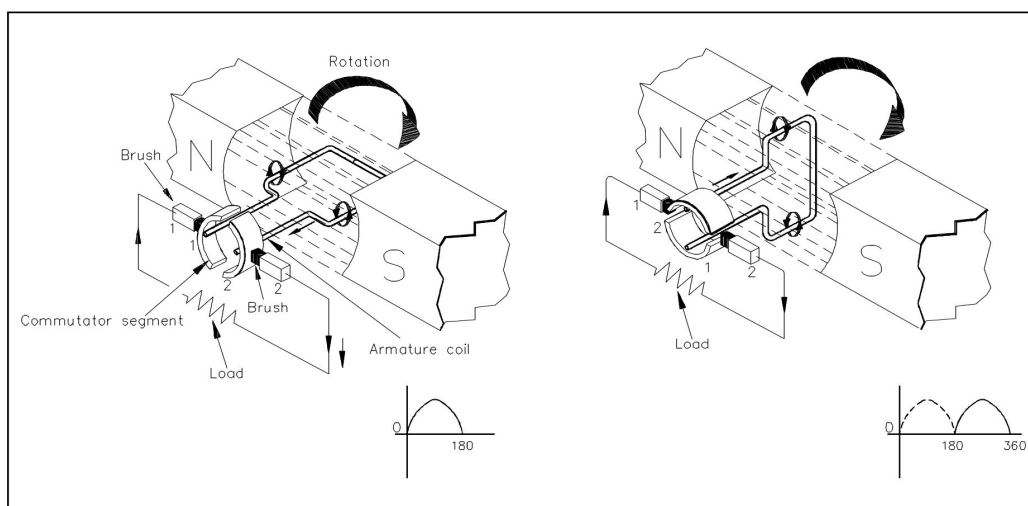


Figura 3. 2 - Funcionamento de um gerador DC

Com base na Fig. 3.2, um condutor em forma de espira é posicionado entre os pólos magnéticos. Enquanto a espira estiver parada, o campo magnético não provoca nela nenhum efeito. No entanto, se a espira for rodada, o fluxo magnético varia na sua área interior. Nesta sequência é induzida uma corrente (lei de Ampere) na espira. A intensidade da corrente induzida depende da intensidade do campo magnético, assim como da velocidade de rotação da espira. Genericamente tem-se:

$$E_g = K\Phi N \quad (3.1)$$

Onde:

E_g - Tensão gerada

K - Constante

Φ - Intensidade do campo magnético

N - Velocidade em rpm

Se se souber o sentido de movimento do condutor e o sentido do campo magnético é possível deduzir, segundo a lei de Ampere, o sentido da corrente eléctrica induzida e portanto a polaridade aos terminais das escovas.

A finalidade do colector é de converter a corrente AC gerada na espira numa corrente rectificada (Fig. 3.2). Serve também como interface de ligação entre as escovas e a espira. Num gerador simples, composto por uma única espira, o colector é feito a partir de duas placas semicilíndricas de um material condutor macio (geralmente cobre).

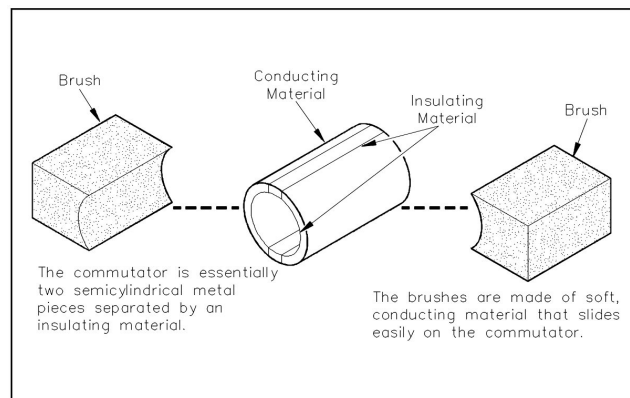


Figura 3.3 - Detalhe do colector e escovas

As placas (ou segmentos) estão separadas através de um material isolante, como mostra a Fig. 3.3. Cada segmento está ligado permanentemente a cada terminação da espira. As escovas, usualmente feitas à base de carbono, são pressionadas através de molas contra o colector de forma a promover o contacto eléctrico. Neste contexto, o colector ao rodar desliza contra as escovas que contactam cada uma das terminações da espira.

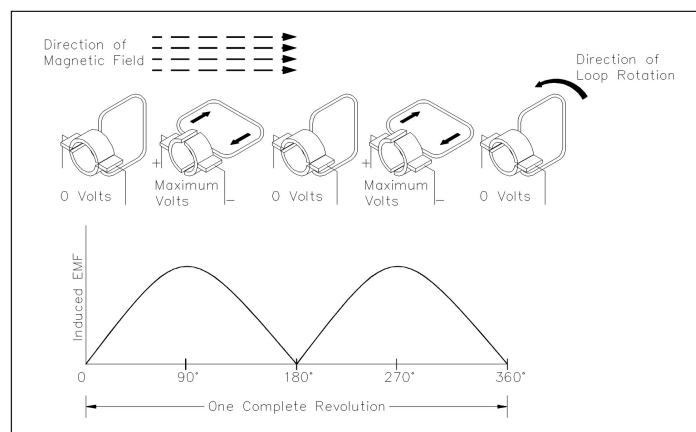


Figura 3. 4 - Acção do comutador

Uma vez que as escovas estão posicionadas em lados opostos no comutador, cada uma vai deslizar ao longo de um segmento e depois ao longo de outro. O conjunto colector/escovas está desenhado de forma a que se verifique uma comutação de segmentos no instante em que a espira inverte a polaridade aos seus terminais. Isto significa que, para um dado sentido de rotação, uma escova é sempre positiva em relação à outra. A tensão observada entre as escovas varia em amplitude entre 0 V e um valor máximo, como demonstra a Fig 3.4. É de salientar que, à medida que as escovas saltam de um segmento para o outro, há um instante em que estas contactam ambos os segmentos adjacentes simultaneamente. A tensão induzida nesse preciso instante é 0 V. Se assim não fosse, surgiriam correntes muito elevadas uma vez que os terminais das espiras estariam curto-circuitados.

3.1.3. Factores de perdas dos geradores DC

De uma maneira geral, existem 4 tipos de factores de perdas que contribuem para a redução da eficiência num gerador DC: perdas no cobre, perdas por corrente de Foucault, perdas por histerese e perdas mecânicas.

Segue-se uma breve descrição da natureza de cada uma delas:

- **Perdas no cobre:** este factor está directamente associado aos enrolamentos de cobre que constituem o indutor e o induzido. Estes condutores possuem uma resistência diferente de 0 e, portanto, dissipam energia por efeito de Joule, i.e., sob a forma de calor na relação $I^2 R$.
- **Perdas por corrente de Foucault:** á medida que o rotor gira no interior do campo magnético, tanto os seus enrolamentos como o núcleo cortam as linhas de fluxo. Uma vez que o núcleo é composto por ferro, uma corrente é-lhe induzida o que se traduz numa circulação de corrente. A estas correntes

que circulam no interior do núcleo de ferro dá-se o nome de “Correntes de Foucault”. Para minimizar este efeito, o núcleo é construído a partir de várias camadas sobrepostas de lâminas de aço isoladas entre si. O isolamento impede que as correntes resultantes passem de uma lâmina para a outra.

- **Perdas por histerese:** Considere-se o rotor como sendo formado por um conjunto de pequenas partículas magnéticas (domínios de Wiess). Uma parte dessas partículas tem o seu domínio magnético alinhado com o campo magnético criado no estator. O número de partículas alinhadas depende da intensidade desse campo. No momento em que o rotor gira, o domínio magnético das partículas alinhadas também roda em relação às partículas que não estão alinhadas, o que provoca uma fricção e, conseqüentemente, calor. Ao calor produzido por esta fricção entre partículas dá-se o nome de perdas por histerese. Para reduzir este tipo de perdas as lâminas que compõem o núcleo do rotor são feitas a partir de um tipo de aço próprio (*silicon steel*). Esse aço sofre várias fases de tempera térmica, cujo processo se denomina de recozimento (*annealing*), e que resulta num material com baixas perdas por histerese.
- **Perdas mecânicas:** as perdas mecânicas estão directamente associadas ao atrito provocado pela fricção dos rolamentos que suportam o eixo do rotor, pela fricção das escovas no colectador e pela fricção do ar no rotor em movimento. Uma limpeza e lubrificação cuidada nos rolamentos assim como uma tensão e assentamento adequados das escovas contribuem para minimizar as perdas mecânicas de um gerador DC.

3.1.4. Excitação do campo magnético

Como já foi referido anteriormente, o campo magnético de um gerador DC pode ser providenciado por electroímãs ou ímãs permanentes. Contudo, é mais comum encontrar máquinas com electroímãs. Para que um gerador DC opere correctamente, o campo magnético tem que ser criado sempre no mesmo sentido e, portanto, a corrente que percorre o electroímã terá de ser DC. Esta corrente de excitação pode ser fornecida de duas formas: através de uma fonte DC externa ou através da própria saída do gerador, sendo este último denominado de “gerador auto-excitado”. Aqui o indutor pode ser ligado em série (Fig. 3.5), em paralelo (Fig. 3.7) ou uma composição de ambas (Fig. 3.9).

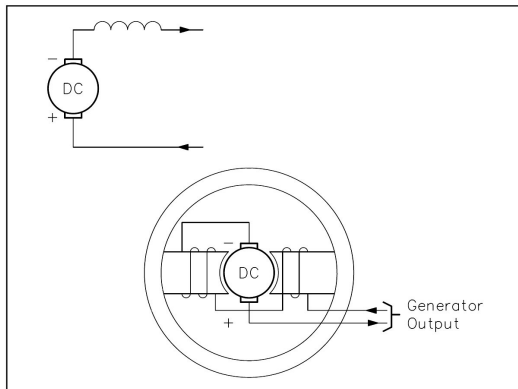


Figura 3. 5 - Esquema excitação série

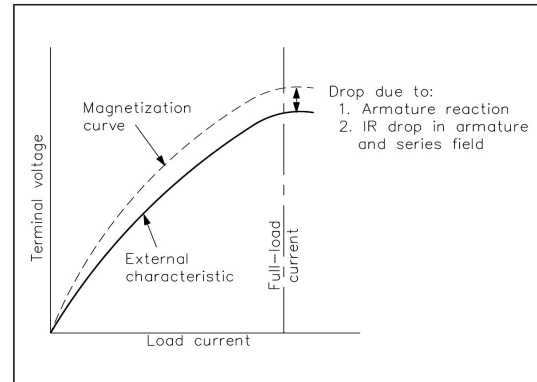


Figura 3. 6 - Curva de carga da excitação série

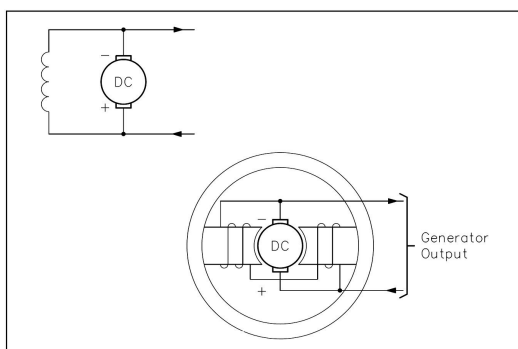


Figura 3. 7 - Excitação paralela

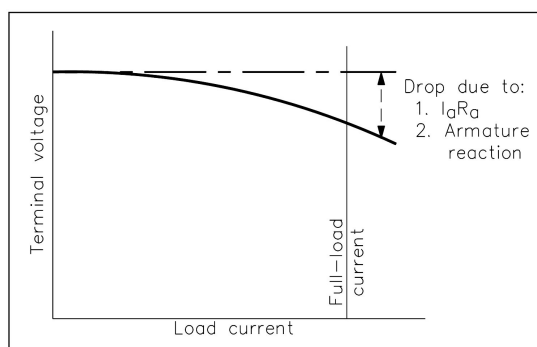


Figura 3. 8 - Curva de carga da excitação paralela

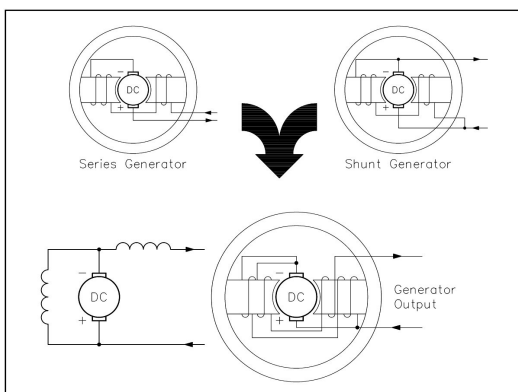


Figura 3. 9 - Excitação composta

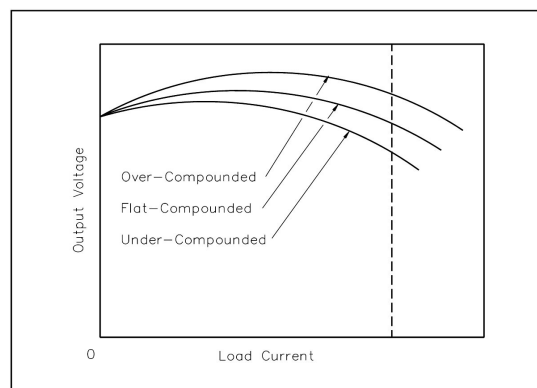


Figura 3. 10 - Curva de carga da excitação composta

De referir que a excitação composta é a que apresenta uma tensão de saída mais estável em função da variação da corrente de carga. A excitação independente, geralmente mais dispendiosa, é usada em aplicações muito específicas onde a tensão de saída deve variar numa ampla gama de valores.

3.1.5. Caracterização dos geradores DC

De um modo geral os geradores DC são avaliados tendo em conta as seguintes especificações:

- **Tensão nominal** (V): esta tensão depende do número de espiras dos enrolamentos assim como no tipo de isolamento usado.
- **Corrente** (A): a secção de fio do cobre e quantidade de calor que a máquina pode dissipar definem a corrente disponível no gerador.
- **Potência** (W): a potência do gerador depende das limitações mecânicas do dispositivo que o faz rodar e das limitações térmicas de condutores, rolamentos e restantes componentes.
- **Velocidade** (rpm): a velocidade máxima é aquela na qual a integridade mecânica da máquina atinge o limiar de resistência e estabilidade.

3.2. Transformadores

3.2.1. O transformador ideal

A lei de Faraday, na sua forma diferencial (Eq. 3.2), diz que o fluxo magnético que atravessa uma indutância produz uma tensão aos seus terminais. Essa tensão é proporcional ao número de espiras e à taxa de variação do fluxo.

$$e = N \frac{d\Phi}{dt} \quad (3.2)$$

Onde:

e - Tensão instantânea aos terminais de uma indutância

N - N° de espiras

Φ - Fluxo magnético

Considere-se então uma indutância em que lhe é aplicada uma tensão variável, que por sua vez, gera um fluxo igualmente variável no tempo. Se na sua proximidade for colocada uma outra indutância de modo a que o fluxo gerado pela primeira passe através da segunda então, de acordo com a Eq. 3.2, irá surgir uma tensão aos terminais da segunda indutância que também será proporcional à taxa de variação do fluxo e ao seu número de espiras.

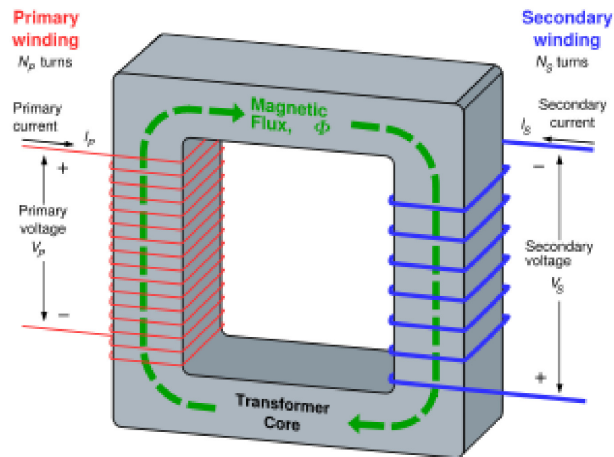


Figura 3. 11 - Princípio do transformador

A Fig. 3.11 mostra que se aplicarmos um circuito magnético fechado (de modo a que todo o fluxo gerado na primeira bobina atravessasse a segunda) pode-se concluir que, considerando a lei de Faraday, a tensão que aparece na segunda bobina será proporcional à tensão aplicada na primeira e à razão entre o número de espiras da primeira sobre a segunda. A partir do momento em que se aplica uma carga aos terminais da segunda bobina, uma corrente irá fluir numa direcção que se opõe a qualquer alteração do fluxo magnético (lei de Lenz). Esta corrente que percorre o segundo enrolamento tende a reduzir o fluxo que atravessa ambas as bobinas. Verifica-se então um abaixamento da impedância vista aos terminais do primário e logo um aumento da corrente. Para a mesma tensão de primário, o fluxo máximo no núcleo atinge valores menores quando comparados com os valores de fluxo na situação em que transformador está em vazio. Uma vez que o fluxo, resultante de qualquer uma das correntes, é proporcional ao produto de uma corrente pelo respectivo número de espiras, obtém-se a relação apresentada na Eq. 3.3. Ao primeiro enrolamento dá-se o nome de primário, e ao segundo de secundário.

$$I_p N_p = I_s N_s \quad (3.3)$$

De forma recíproca tem-se que:

$$\frac{V_s}{N_s} = \frac{V_p}{N_p} \quad (3.4)$$

Dividindo a Eq. 3.3 pela Eq. 3.4 conclui-se que:

$$\frac{V_s}{I_s} = Z_s = \left(\frac{N_s}{N_p} \right)^2 Z_p \quad (3.5)$$

Onde:

I_p e I_s - Correntes no primário e secundário respectivamente

N_p e N_s - N° de espiras do primário e secundário respectivamente

V_p e V_s - Tensões no primário e secundário respectivamente

Z_p e Z_s - Impedâncias vistas do primário e secundário respectivamente

As considerações anteriormente expostas assim como as equações resultantes descrevem, de forma resumida, o funcionamento básico de um transformador ideal.

3.2.2. Modelo equivalente do transformador

O design e utilização de transformadores requerem frequentemente que se tenha em consideração as suas características não ideais. Daí que muitas vezes se recorra à análise do modelo equivalente do transformador real. É importante realçar o facto dos modelos equivalentes não serem réplicas exactas mas sim aproximações convenientes daquilo que se pretende representar.

Segue-se uma breve síntese dos vários elementos que compõem o modelo equivalente completo do transformador real.

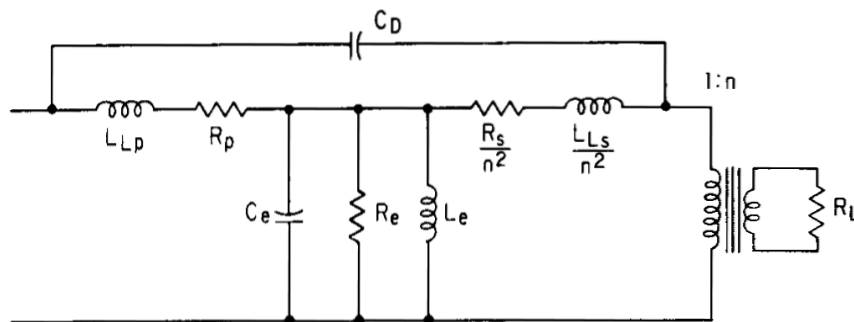


Figura 3. 12 - Modelo equivalente do transformador real

Existe uma corrente de magnetização do lado do primário que é responsável pela criação do fluxo num transformador. Esta corrente é independente da corrente de carga transformada e não provoca uma perda de energia. Daí que seja representada na Fig. 3.12 por uma indutância, L_e , em paralelo com um transformador ideal. O valor desta indutância depende da natureza dos materiais ferromagnéticos usados na construção do núcleo. Por outro lado, esses materiais ferromagnéticos são responsáveis pelas chamadas perdas no núcleo, que não são mais do que uma combinação entre as perdas

por histerese e as perdas por correntes de Foucault (como se verá mais adiante) e é representada no esquema por uma resistência, R_e , em paralelo. Uma outra característica importante a considerar tem a ver com o facto de ambos os enrolamentos serem constituídos por condutores que possuem uma resistividade finita. Assim, cada enrolamento terá de ter a sua resistência série equivalente (representadas por R_p e R_s). Uma vez que R_s está reflectida no lado do primário, o seu valor vem dividido pela relação do número de espiras ao quadrado (Eq. 3.5). A resistência série do primário (R_p) é atravessada pela soma de todas as correntes anteriores já referidas pelo que esta se apresenta antes dos elementos em paralelo. Ambos os enrolamentos são construídos a partir de múltiplas espiras de fio de cobre enroladas em torno do núcleo. As espiras são mantidas muito próximas entre si, o que provoca com que haja uma capacidade parasita distribuída ao longo de todo o conjunto. Esta capacidade distribuída pode ser traduzida por um condensador em série e outro em paralelo, C_d e C_e respectivamente. O modelo do transformador ideal assume que o fluxo magnético atravessa todas as espiras de ambos os enrolamentos. Na prática existe uma pequena parte desse fluxo que se perde, situação que é observada quando se detecta uma pequena indutância no primário, no momento em que o secundário está curto-circuitado. Este factor sugere para o circuito equivalente duas indutâncias série chamadas indutância de fuga (L_{Lp} e L_{Ls}). A indutância está reflectida no primário por L_{Ls}/n^2 . Dependendo do tipo de aplicação, este modelo equivalente completo do transformador pode repartir-se em diversos circuitos mais simples para análise.

3.2.3. O uso de materiais ferromagnéticos em transformadores

As características que definem um transformador ideal podem ser traduzidas em termos de especificações de materiais que compõem o núcleo ideal. Contudo, o afastamento dos transformadores reais em relação ao modelo ideal reflecte em larga medida o afastamento dos materiais usados na prática em relação às propriedades ideais para o núcleo. De uma maneira geral os materiais magnéticos com as melhores propriedades são também os mais dispendiosos em termos económicos. Por essa razão deve-se escolher o material com a performance mais baixa que se satisfaça as especificações mais importantes para um dado transformador. Os materiais ferromagnéticos possuem um conjunto de propriedades cujas características são descritas através de uma curva de magnetização. A curva de magnetização é um gráfico que representa a densidade de fluxo magnético em função da força de magnetização.

$$B = \mu H \quad (3.6)$$

Onde:

B - Densidade de fluxo (ou indução)

μ - Permeabilidade magnética

H - Força de magnetização (ou campo magnético)

Ambos estão relacionados pela Eq. 3.6, em que μ é a permeabilidade magnética, propriedade associada ao meio. A permeabilidade do vazio é 1. Para os materiais ferromagnéticos a permeabilidade (quando a densidade de fluxo é baixa) é várias vezes a do vazio. À medida que a densidade de fluxo aumenta, o valor da permeabilidade sobe acima do seu valor inicial e depois diminui, aproximando-se de 1 na região do gráfico denominada “saturação”. Se a força de magnetização decresce após uma trajetória de subida, verifica-se que a densidade de fluxo assume valores superiores àqueles que se observaram inicialmente, aquando da subida do campo H. Quando a força de magnetização se anula, ainda existe um determinado valor remanescente de densidade de fluxo (magnetismo residual). É necessário reverter o sentido do campo H de modo a tornar a anular o valor do fluxo. Este comportamento é análogo para as regiões negativas da curva.

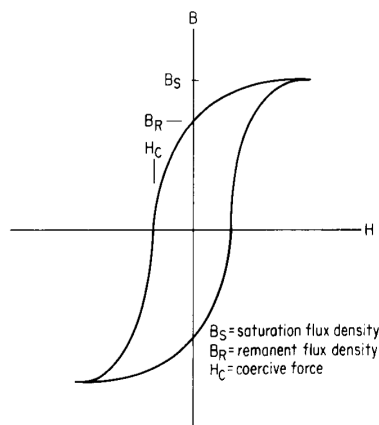


Figura 3. 13 - Curva de histerese

Toda esta sequência descrita denomina-se “curva de histerese”, conforme se pode observar na Fig. 3.13. A forma desta curva caracteriza os diversos materiais ferromagnéticos usados na construção do núcleo e depende do tipo de material, da sua geometria, da amplitude e frequência de força da magnetização. A sua área interior corresponde às perdas magnéticas, assunto que será discutido mais adiante.

As propriedades dos materiais ferromagnéticos dependem do tipo de composição do próprio material, da forma como é fabricado e do tratamento térmico a que é sujeito. O ferro, o níquel e o cobalto são os elementos mais usados no seu fabrico. Os materiais ferromagnéticos dividem-se em duas classes: materiais duros e macios. Os materiais duros têm uma força coerciva e um fluxo remanescente elevados (H_C e B_R respectivamente, representados na Fig. 3.13). São usados no fabrico de ímanes permanentes. Por outro lado os materiais macios, com uma força coerciva mais reduzida, permitem a aplicação de campos alternados o que os torna ideais para a construção de

transformadores. Segue-se um resumo dos principais materiais macios utilizados assim como as suas características mais relevantes.

- **Liga de aço e silício** (*silicon steel alloy*): é de longe o material mais usado para a construção de núcleos de transformadores. É composto por aço e cerca de 0,5% a 3,25% de silício, o que permite um acréscimo na permeabilidade magnética e uma diminuição nas perdas por histerese. É laminada em folhas finas com isolamento eléctrico que por sua vez são sobrepostas, criando um núcleo com menores perdas por correntes de Foucault. A sua aplicação centra-se em transformadores de potência e de baixa frequência. Folhas de calibre mais fino permitem estender o seu uso até á gama de frequências áudio. Por outro lado a magnetização assimétrica e o *air gap* no circuito magnético provocam uma menor variação do valor efectivo de permeabilidade quando comparado com outros materiais.

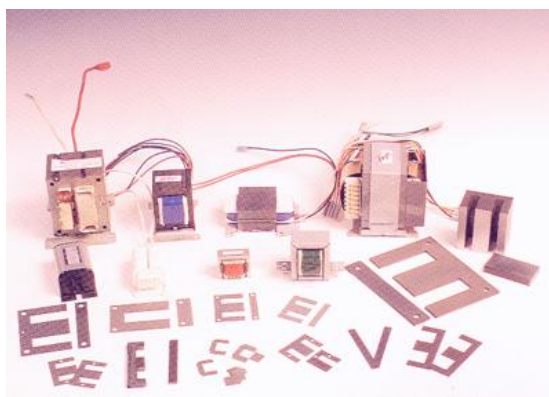


Figura 3. 14 - Exemplo de alguns núcleos de ferro

- **Liga de ferro e níquel** (*niquel_iron alloy*): esta liga possui uma permeabilidade superior à do aço e silício mas satura para valores de densidade de fluxo mais baixos. É usada sobretudo em aplicações que exigem uma maior largura de banda como é o caso de circuitos de alta performance que funcionam na banda de áudio e vídeo.
- **Liga de ferro e cobalto** (*cobalt-iron alloy*): este material satura para valores de fluxo mais elevados do que qualquer outro no mercado. Porém, o custo e escassez do cobalto limitam a importância comercial desta liga.
- **Liga de vidro e metal magnético** (*magnetic metallic glasses*): esta liga é composta por ferro, boro e silício. Estes compostos são arrefecidos muito rapidamente durante a fase de moldagem do núcleo, o que impede a formação de cristais magnéticos mais comuns (como os que se formam nos núcleos de ferro). Neste contexto, este novo material evidencia características

como elevada resistividade (e portanto menores perdas por correntes de Foucault) e elevado nível de saturação (superior aos dos núcleos de ferrite) o que o torna indicado aplicações de alta-frequência. Por se tratar de um material relativamente recente, a sua disponibilidade no mercado está limitada a apenas algumas formas geométricas para núcleos.

- **Liga de pó magnético** (*powdered magnetic alloy*): Este material é usado em aplicações que exigem um elevado factor de qualidade (factor Q em que $Q = \omega L/R$). A sua permeabilidade é inferior à das ligas metálicas no estado sólido ou ferrites, no entanto, o *air gap* efectivo resultante da compressão do pó (o *air gap* fica mais largo) torna estes núcleos mais tolerantes a magnetizações DC.
- **Ferrites**: as ferrites são uma classe importante de materiais magnéticos, cuja aplicação se estende desde a gama áudio até à ordem dos Mhz. A sua composição baseia-se em óxido de ferro ligado com outros óxidos (magnésio, zinco e níquel), envolvendo um processo cerâmico. Um dos grandes trunfos dos núcleos de ferrite reside no facto de possuírem uma resistividade elevada. Isto é uma vantagem decisiva sobre as ligas metálicas a alta-frequência uma vez que as perdas por correntes de Foucault são fortemente minimizadas. A principal desvantagem está relacionada com a natureza do material em si assim como o processo de fabrico que, no conjunto, restringem o tamanho máximo que os núcleos de ferrite podem ter. Uma outra limitação está relacionada com a temperatura de Curie. A temperatura de Curie é aquela acima da qual o material perde as suas propriedades magnéticas que, no caso das ferrites, é mais baixo (entre 100 e 300 C°) quando comparado com outras ligas.

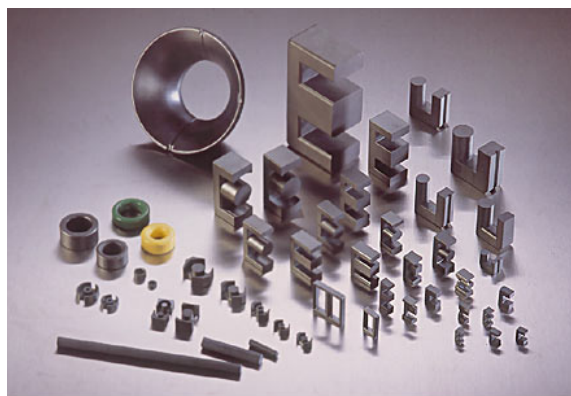


Figura 3. 15 - Exemplo de alguns núcleos de Ferrite

No anexo D (pág. 102) encontra-se uma tabela com as propriedades típicas dos materiais magnéticos macios mais importantes.

3.2.4. Perdas no núcleo

As perdas no núcleo estão relacionadas com a energia dissipada nesse elemento. Existem dois tipos de perda: perdas por histerese e perdas por correntes de Foucault.

As perdas por histerese reflectem a energia gasta para alinhar e rodar as partículas magnéticas elementares do material que compõe o núcleo, à semelhança do que se passa nos geradores.

$$P = \frac{fV \times 10^{-8}}{0.4\pi} \int H \cdot dB \quad (3.7)$$

Onde:

B - Indução

H – Campo magnético

P - Potência dissipada

f - Frequência

V - Volume do núcleo (cm³)

A Eq. 3.7 quantifica de forma aproximada as perdas por histerese de um núcleo genérico e que foi deduzida em [1]. Um resultado importante a reter é que a potência dissipada, nestas circunstâncias, é directamente proporcional à frequência de funcionamento do transformador. O valor do integral não é mais do que a área da curva de histerese.

As perdas por correntes de Foucault são causadas por correntes que circulam num núcleo formado por material condutor. A partir do momento em que se varia o fluxo que atravessa o núcleo, são induzidas tensões no seu interior, e portanto, correntes que formam padrões circulares. Estas correntes acabam por produzir calor no material.

$$P_{av} = \frac{K^2 W^2 B_{max}^2 f}{16\rho(K^2 + 1)\tau} \times 10^{-16} \quad (3.8)$$

Onde:

P_{av} - Potência média dissipada

K - Factor adimensional relacionado com a geometria do núcleo

W - Largura de uma secção transversal do núcleo (cm)

B_{max} - Densidade de fluxo máxima (Gauss)

f - Frequência

ρ - Resistividade do material por unidade de volume (Ω/cm³)

τ - Largura do pulso

A Eq. 3.8 (deduzida em [1]) é uma aproximação da potência dissipada num transformador por correntes de Foucault, na situação em que é excitado com uma onda quadrada. Também se verifica uma dependência directa com a frequência, à semelhança do resultado anterior.

3.2.5. O efeito dos materiais do núcleo na largura de banda

As propriedades dos materiais que compõem o núcleo determinam a largura de banda de um transformador. Deve-se olhar para o transformador como um filtro passa-banda. A frequência de corte mais baixa depende da indutância paralela (L_e na Fig. 3.12) que por sua vez depende da permeabilidade do material, do número de espiras do primário e da geometria do núcleo. A frequência de corte superior é determinada pela indutância de fugas e pela capacidade distribuída que estão condicionadas pelo número de espiras, pela geometria e volume dos enrolamentos primário e secundário. Para alcançar uma largura de banda mais ampla é necessário um núcleo com elevada permeabilidade (que se traduz num valor de L_e superior) aliado a um menor número de espiras possível (que tem implicações na frequência de corte superior). À medida que a frequência diminui, a densidade de fluxo aumenta para um valor de tensão fixo. Por outro lado a permeabilidade do núcleo diminui à medida que nos aproximamos do ponto de saturação. Esta situação impõe um requisito adicional do material magnético: deve-se escolher materiais com valores de saturação da densidade de fluxo superiores, para se obter uma boa resposta às baixas frequências.

3.3. Topologias de Inversores e Suas Características

3.3.1. Topologias de inversores

Um inversor é um dispositivo que converte a energia eléctrica de DC para AC por intermédio de elementos de comutação rápidos. Trata-se de um conversor que está incluído uma classe específica dentro do universo das fontes comutadas. As fontes comutadas são dispositivos de regulação (de tensão ou corrente) cujo funcionamento se baseia em topologias de circuitos com *switches* (transístores bipolares, MOSFET's, TRIAC's, IGBT's) e elementos que armazenam energia (indutâncias e condensadores). Caracterizam-se por terem uma maior eficiência e melhor relação peso/potência quando comparados com os reguladores lineares mas apresentam um maior grau de complexidade.

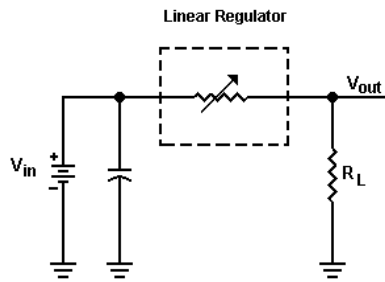


Figura 3. 16 - Regulador linear

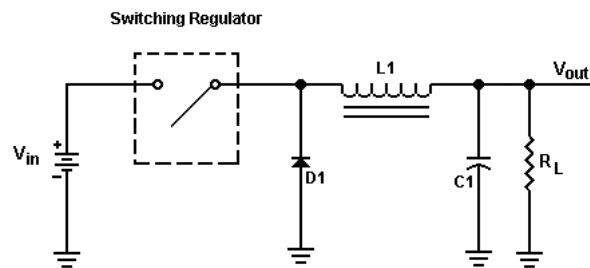


Figura 3. 17 - Regulador comutado

Para entender melhor o funcionamento dos inversores segue-se uma apresentação das várias topologias usadas para efeito.

- **Conversor *Flyback* com transformador de isolamento**

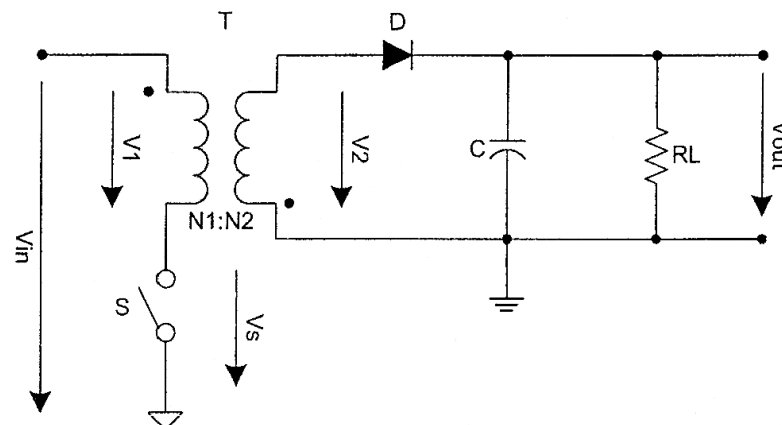


Figura 3. 18 - Conversor *Flyback* com transformador de isolamento

Esta topologia, apesar de não ser um inversor que produz uma saída AC, serve de ponto de partida para as topologias dos inversores propriamente ditos. Caracteriza-se por usar um transformador que providencia saídas isoladas. Este tipo de circuito converte uma tensão DC de entrada V_{in} numa tensão de saída V_{out} regulada, cujo valor depende do *duty-cycle* do sinal de controlo do interruptor assim como da relação de transformação de T (Eq. 3.9).

$$V_{out} = V_{in} \left(\frac{N_2}{N_1} \right) \frac{t_{on}}{t_{off}} \quad (3.9)$$

Onde:

V_{in} - Tensão de entrada

V_{out} - Tensão de saída

N_2 - N° de espiras do secundário
 N_1 - N° de espiras do primário
 t_{on} - Tempo do comutador ligado
 t_{off} - Tempo do comutador desligado

No momento em que S fecha (durante o intervalo de tempo igual a t_{on}) a corrente no primário do transformador T sobe linearmente com um declive Δi_L^+ igual a V_{in}/L_{prim} . Durante este intervalo de tempo, devido à relação de fase entre os enrolamentos do transformador, o diodo D está inversamente polarizado e a corrente na carga RL é fornecida pela energia armazenada no condensador C. Quando o inversor S abre (durante um período de tempo t_{off}) a tensão no secundário inverte a polaridade (isto porque a corrente que atravessa a indutância do secundário não pode variar instantaneamente), polarizando directamente o diodo D. Durante T_{off} a corrente no secundário de T decresce linearmente com uma inclinação Δi_L^- através do diodo D. Toda esta dinâmica encontra-se representada nas formas de onda da Fig. 3.19.

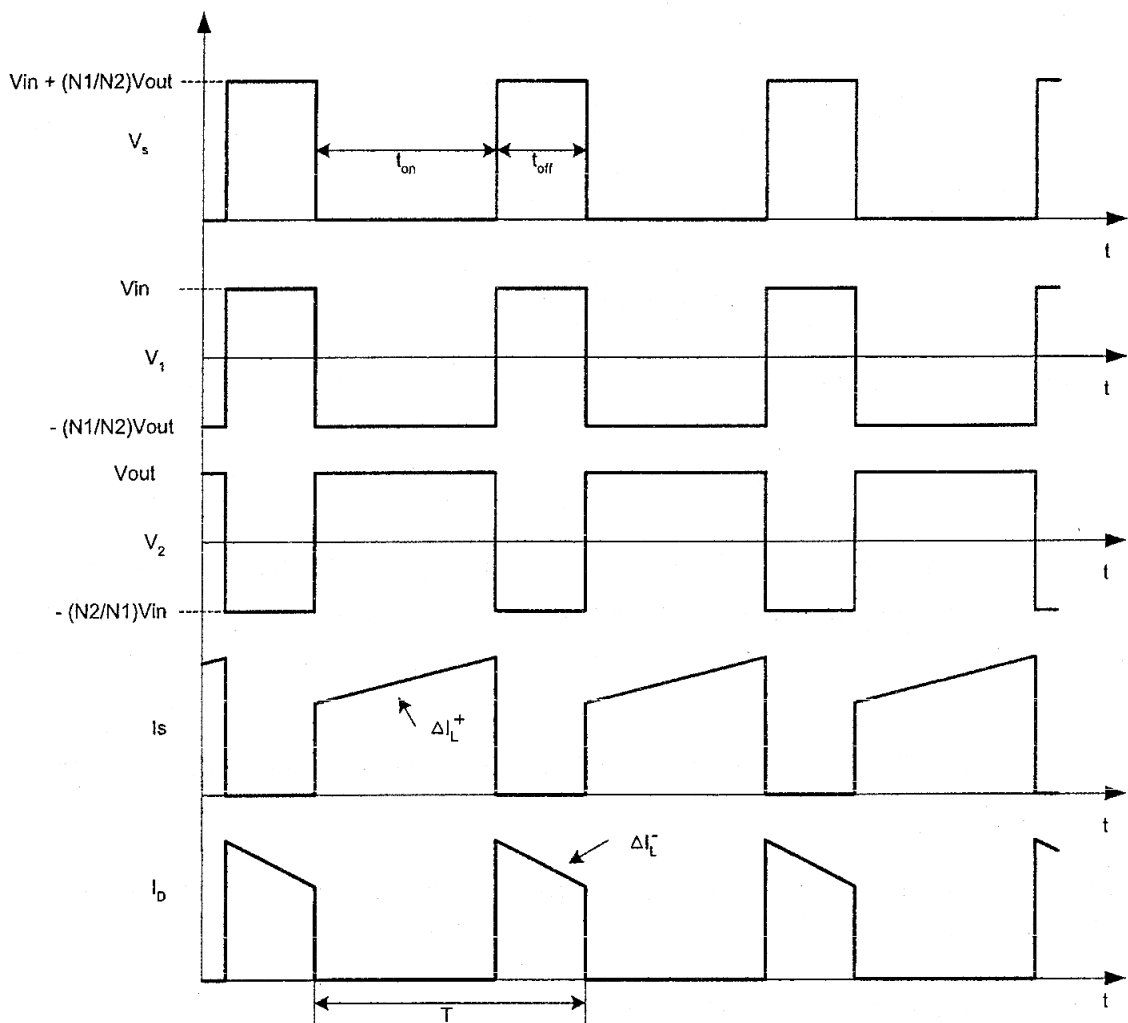


Figura 3. 19 - Principais formas de onda do conversor *Flyback*

Apesar de ser uma topologia simples, apresenta o inconveniente do interruptor S magnetizar o transformador T numa única direcção. Como consequência, T acumula magnetização remanescente e poderá atingir o ponto de saturação mais rapidamente com correntes superiores.

- **Inversor *Half-Bridge***

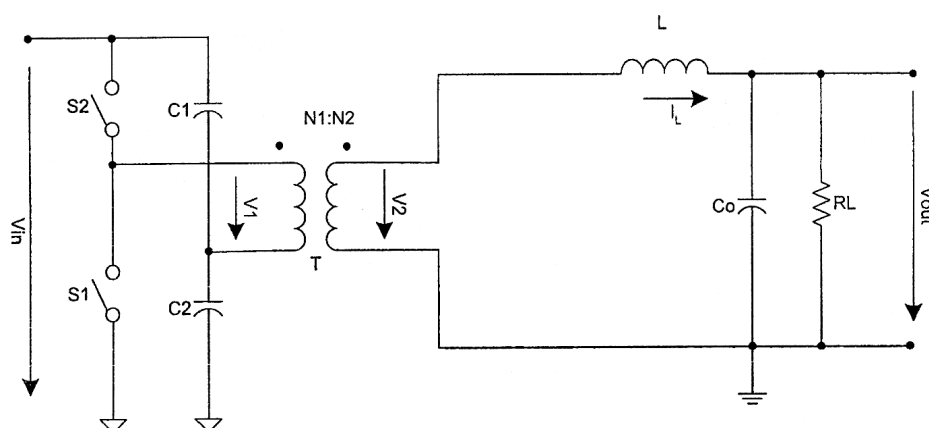


Figura 3. 20 - Inversor *Half-Bridge*

A Fig. 3.20 apresenta o esquema de um inversor *half-bridge* com transformador de isolamento. Ao contrário da topologia anterior, este circuito magnetiza o transformador nos dois sentidos. Os interruptores S1 e S2 comutam alternadamente o que faz com que surja no secundário uma tensão AC. No ponto médio entre os condensadores a tensão é igual a $V_{in}/2$. Isto significa que esta topologia só aproveita metade da tensão de entrada. Trata-se de uma desvantagem pois obriga a que o transformador tenha uma relação de espiras maior para compensar essa perda de amplitude. Um maior número de espiras implica um transformador mais caro, mas sobretudo, um transformador com largura de banda mais limitada, especialmente às altas-frequências.

Para este circuito a tensão de saída será:

$$V_{out} = +\frac{1}{2}V_{in} \frac{N_2}{N_1} \frac{t_{2on}}{T} \quad \text{se S2 on e S1 off}$$

$$V_{out} = -\frac{1}{2}V_{in} \frac{N_2}{N_1} \frac{t_{1on}}{T} \quad \text{se S1 on e S2 off}$$
(3.10)

Onde:

T - Período de comutação

Um outro inconveniente associado a este circuito tem a ver com o uso de um interruptor no lado positivo de V_{in} (*high-side* S2) e outro no lado negativo (*low-side* S1). Isto implica o uso de transístores de natureza diferente (por exemplo, um MOSFET de canal P para S2 e um MOSFET de canal N para S1) e pode provocar uma assimetria na dinâmica de comutação alternada, isto é, caso um dos transístores conduza mais que o outro, em média, será adicionada uma componente DC indesejável na tensão de saída.

- **Inversor *Full-Bridge***

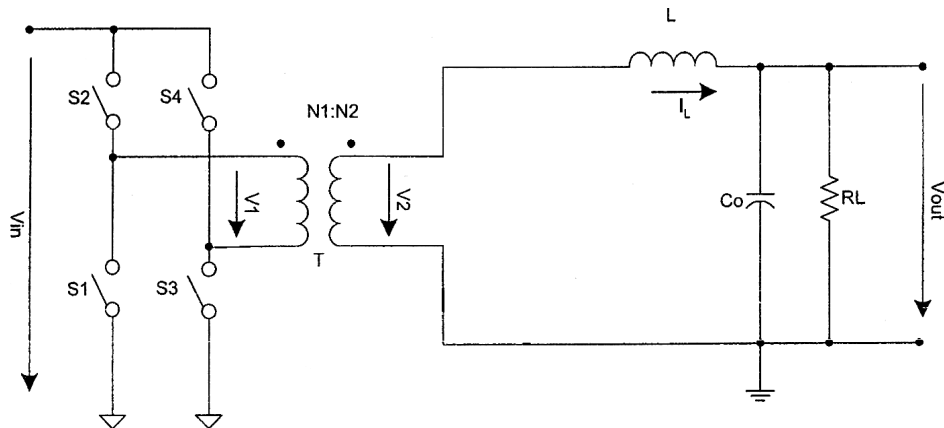


Figura 3. 21 - Inversor *Full-Bridge*

A Fig. 3.21 mostra o circuito de um inversor *full-bridge* com transformador de isolamento. A principal diferença em relação à topologia anterior está na presença de quatro elementos de comutação. Desta forma aproveita-se a totalidade da amplitude de V_{in} para gerar uma tensão alternada. Como consequência o transformador desta topologia precisará de metade do número de espiras no primário relativamente ao transformador do inversor *half-bridge*, para a mesma tensão de entrada, com todas as vantagens que isso implica. Contudo, o uso de quatro comutadores aumenta a complexidade do circuito de controlo associado incidindo directamente no aumento do custo final de uma aplicação baseada neste esquema. Tipicamente os inversores de alta potência que trabalhem a frequências superiores baseiam-se na topologia *full-bridge*.

A expressão da tensão de saída para este circuito será:

$$V_{out} = +V_{in} \frac{N_2}{N_1} \frac{t_{2on}}{T} \quad \text{se S2, S3 on e S1, S4 off}$$

$$V_{out} = -V_{in} \frac{N_2}{N_1} \frac{t_{1on}}{T} \quad \text{se S1, S4 on e S2, S3 off}$$
(3.11)

- **Inversor *Push-Pull***

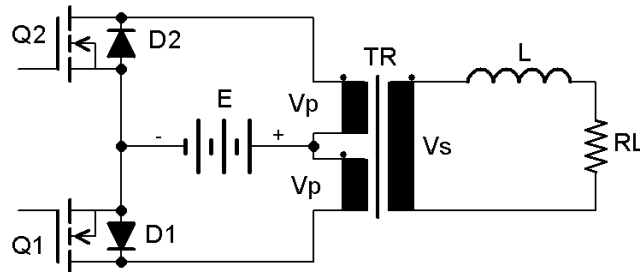


Figura 3. 22 - Inversor *Push-Pull*

O esquema da Fig. 3.22 representa um inversor *push-pull* com transformador de isolamento e é nesta topologia que se baseia este trabalho. Este circuito segue uma linha base semelhante aos anteriores, no entanto, possui um transformador com tomada central no primário o que permite a inclusão de dois comutadores referenciados ao mesmo potencial (por exemplo, dois transístores MOSFET de canal N referenciados à massa, i.e., *low-side*). Esta característica evita o problema já referido para os casos do *half-bridge* e *full-bridge*. A tensão DC de entrada é aplicada alternadamente entre a tomada central e cada uma das terminações do primário. Isto significa que a corrente flui apenas numa das metades do enrolamento primário em cada meio ciclo do período da tensão AC gerada. Trata-se de uma clara desvantagem na medida em que o aproveitamento do cobre do transformador é mais pobre quando comparado com as outras duas topologias. No entanto, por se tratar de um circuito simples, o *push-pull* é largamente usado em inversores de baixa potência sobretudo em modo *stand-alone*. Para aplicações de maior potência (na ordem dos KW) o *push-pull* deixa de ser apropriado, porque implica o uso de um transformador demasiado grande e isso compromete o custo final assim como a relação peso/potência.

3.3.2. Princípio de funcionamento do inversor *Push-Pull*

Segue-se uma descrição resumida do comportamento dinâmico do *push-pull*, assim como as várias formas de onda associadas.

- **Intervalo $[t_0, t_1]$ Fig. 3.23 a):**

Etapa de transferência de energia. O transístor Q1 começa a conduzir; tanto no primário como no secundário a corrente cresce até atingir um valor máximo (I_p e I_s

respectivamente) que depende da tensão de entrada, da relação de transformação e da carga. A tensão no secundário é igual a $V_s = (n_s/n_p) V_p$ onde $V_p = +E$.

- **Intervalo $[t_1, t_2]$** Fig. 3.23 b):

O transistor Q1 abre; nesta fase a corrente do secundário I_s não se inverte imediatamente e, pela lei de Lenz, existe uma inversão instantânea da tensão da indutância L. O diodo D2 fica directamente polarizado e começa a conduzir. As correntes no secundário e no primário decrescem até se anularem. A tensão no secundário é igual a $V_s = (n_s/n_p) V_p$ onde $V_p = -E$.

- **Intervalo $[t_2, t_3]$** Fig. 3.23 c):

Ambos os transistores estão em aberto. As tensões e correntes no transformador são nulos.

- **Intervalo $[t_3, t_4]$** Fig. 3.23 d):

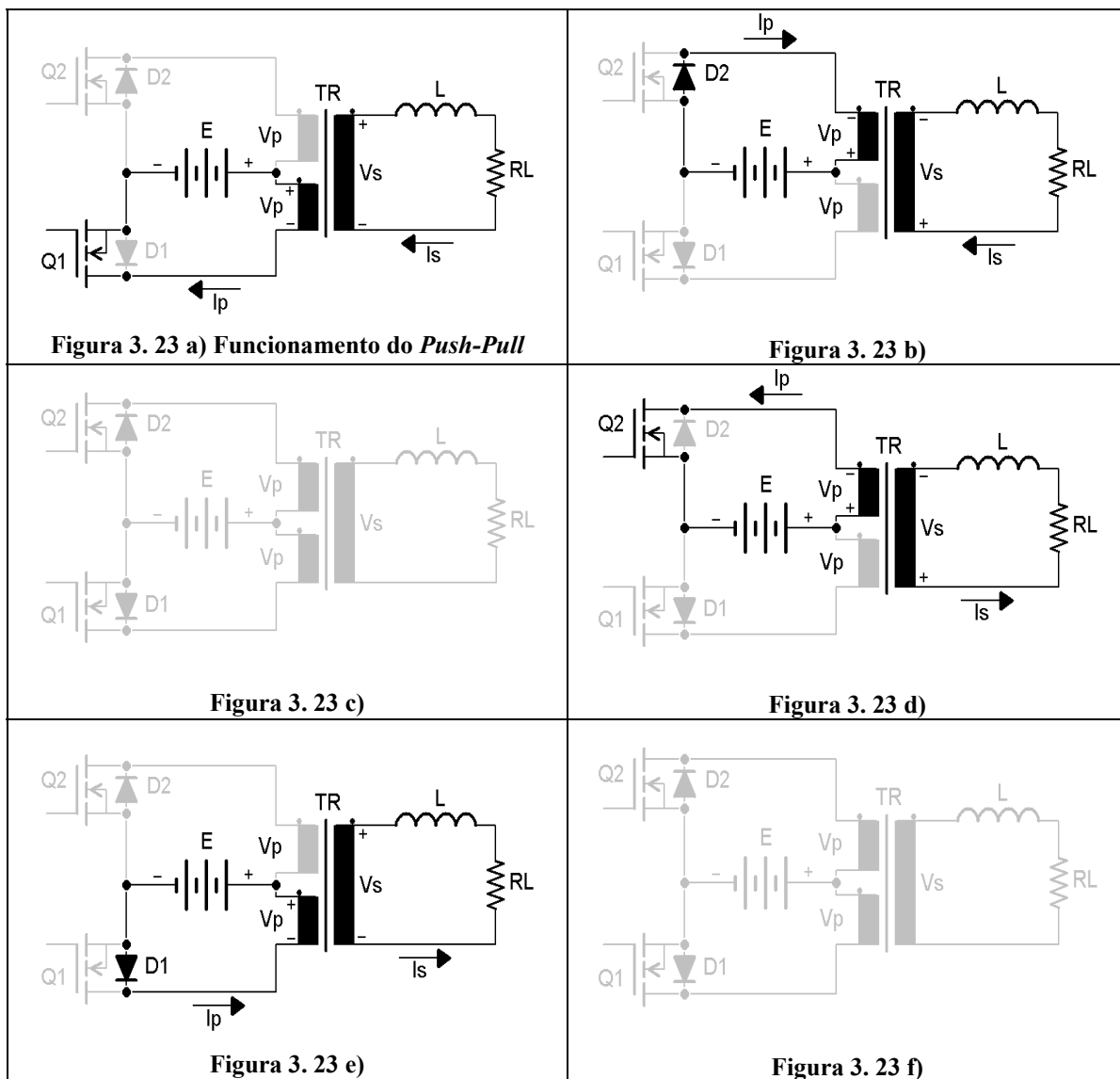
Etapa de transferência de energia. O transistor Q2 começa a conduzir; tanto no primário como no secundário a corrente cresce até atingir um valor máximo (I_p e $-I_s$ respectivamente) que depende da tensão de entrada, da relação de transformação e da carga. A tensão no secundário é igual a $V_s = (n_s/n_p) V_p$ onde $V_p = -E$.

- **Intervalo $[t_4, t_5]$** Fig. 3.23 e):

O transistor Q2 abre; nesta fase a corrente do secundário I_s não se inverte imediatamente e, pela lei de Lenz, existe uma inversão instantânea da tensão da indutância L. O diodo D1 fica directamente polarizado e começa a conduzir. As correntes no secundário e no primário decrescem até se anularem. A tensão no secundário é igual a $V_s = (n_s/n_p) V_p$ onde $V_p = +E$.

- **Intervalo $[t_5, t_6]$** Fig. 3.23 f):

Ambos os interruptores estão em aberto. As tensões e corrente no transformador são nulos.



A Fig. 3.24 mostra as principais formas de onda observáveis nos vários pontos do circuito, onde:

- (A) – Forma de onda da tensão (a preto) e corrente (a vermelho) no secundário de TR.
- (B) – Forma de onda da tensão (a preto) e corrente (a vermelho) no primário de TR.
- (C) – Forma de onda da tensão (a preto) e corrente (a vermelho) entre o dreno e a fonte de Q1.
- (D) – Forma de onda da tensão (a preto) e corrente (a vermelho) aos terminais de D1.
- (E) – Pulsos aplicados às gates de Q1 e Q2.

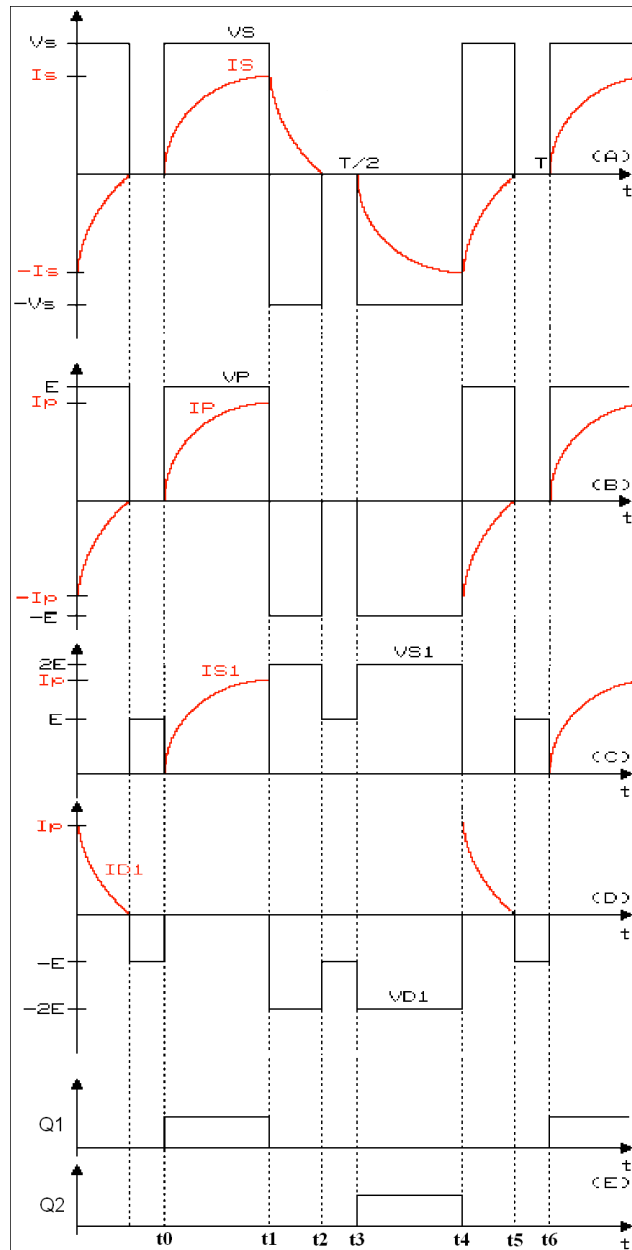


Figura 3. 24 - Formas de onda do *Push-Pull*

3.3.3. Análise da tensão de entrada e saída no transformador

O objectivo deste ponto é quantificar em função do tempo a forma de onda que é aplicada no transformador da Fig. 3.22 assim como a saída resultante.

- Tensão de entrada

Considere-se a seguinte forma de onda para a tensão de entrada dos enrolamentos do primário:

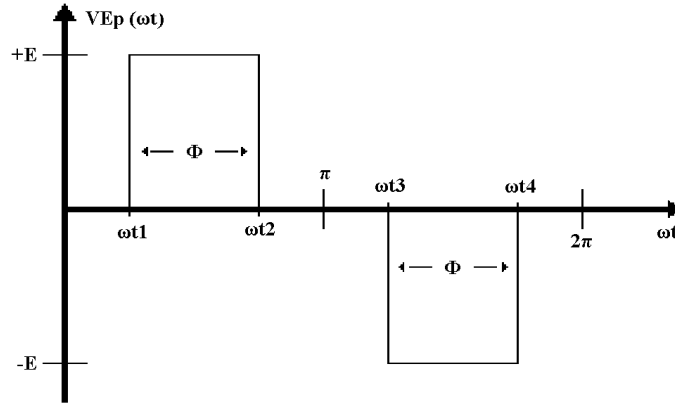


Figura 3. 25 - Forma de onda da tensão de entrada

Decompondo a função da Fig. 3.25 em termos de série de Fourier, tem-se:

$$V_{EP}(\omega t) = \sum_{n=0}^{\alpha} [a_n \cos(n\omega t) + b_n \sin(n\omega t)] \quad (3.12)$$

Onde os respectivos coeficientes são:

$$a_n = \frac{1}{\pi} \int_0^{2\pi} V_{EP}(\omega t) \cos(n\omega t) d(\omega t) \quad (3.13)$$

$$b_n = \frac{1}{\pi} \int_0^{2\pi} V_{EP}(\omega t) \sin(n\omega t) d(\omega t) \quad (3.14)$$

Como $V_{EP}(\omega t)$ é uma função ímpar, os coeficientes a_n são todos nulos. Considerando também que a função tem simetria de um quarto de onda, só existem os termos b_n , para todo o n ímpar. Deste modo é suficiente integrar no intervalo $[0 \pi]$.

$$b_n = \frac{2}{\pi} \int_0^{\pi} V_{EP}(\omega t) \sin(n\omega t) d(\omega t) \quad (3.15)$$

Para $\omega t_1 \leq \omega t \leq \omega t_2$ $V_{EP}(\omega t) = E$ e $V_{EP}(\omega t) = 0$ fora desse intervalo. Então:

$$b_n = \frac{2}{\pi} \int_{\omega t_1}^{\omega t_2} E \sin(n\omega t) d(\omega t) \quad (3.16)$$

Uma vez que $\omega t_1 = (\pi - \Phi)/2$ e $\omega t_2 = (\pi + \Phi)/2$, substituindo vem:

$$\begin{aligned}
b_n &= \frac{2}{\pi} \int_{\frac{\pi-\phi}{2}}^{\frac{\pi+\phi}{2}} E \sin(n\omega t) d(\omega t) \Leftrightarrow \\
\Leftrightarrow b_n &= j^{(n-1)} \frac{4E}{n\pi} \sin\left(\frac{n\phi}{2}\right) \\
j &= \sqrt{-1}; n = 1, 3, 5, 7, \dots
\end{aligned} \tag{3.17}$$

Portanto, a expressão em série de Fourier da forma de onda da tensão de entrada no transformador do inversor Push-Pull é dada por:

$$V_{EP}(\omega t)_n = \frac{4E}{\pi} \sum_{n=1,3,5,\dots}^{\alpha} j^{(n-1)} \frac{\sin\left(\frac{n\phi}{2}\right)}{n} \sin(n\omega t) \tag{3.18}$$

Para $n = 1$ (harmónico fundamental) tem-se:

$$V_{EP}(\omega t)_1 = \frac{4E}{\pi} \sin\left(\frac{\phi}{2}\right) \sin(\omega t) \tag{3.19}$$

O valor máximo da função é dado por:

$$V_{EP} \max = \frac{4E}{\pi} \sin\left(\frac{\phi}{2}\right) \tag{3.20}$$

- **Tensão de saída**

Considere-se a seguinte forma de onda para a tensão de saída do enrolamento secundário:

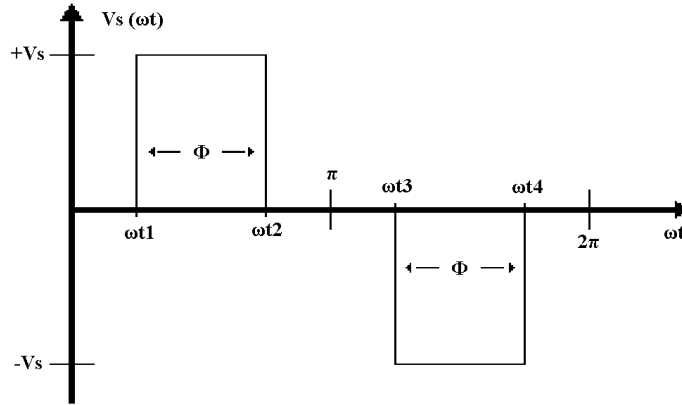


Figura 3. 26 - Forma de onda da tensão de saída

Seguindo o mesmo raciocínio do ponto anterior, a expressão para a série de Fourier da tensão de saída é dada por:

$$V_s(\omega t)_n = \frac{4V_s}{\pi} \sum_{n=1,3,5,\dots}^{\infty} j^{(n-1)} \frac{\sin\left(\frac{n\phi}{2}\right)}{n} \sin(n\omega t) \quad (3.21)$$

O valor máximo da função para $n = 1$, sabendo que $N_s/N_p = V_s/V_p$ e $V_p = E$, é:

$$V_s \max = \frac{4EN_s}{\pi N_p} E \sin\left(\frac{\phi}{2}\right) \quad (3.22)$$

3.3.4. Técnicas de modulação para inversores

As principais estratégias usadas para modelar uma saída AC num inversor *push-pull* são as seguintes: função onda quadrada (*square wave*), função seno modificado (*modified sinewave*) e modulação do seno por largura de pulso (SPWM). As duas primeiras são técnicas muito simples (Fig. 3.27) e são bastante usadas em inversores *stand-alone* de baixo custo que alimentam aparelhos onde a forma de onda de alimentação não é crítica.

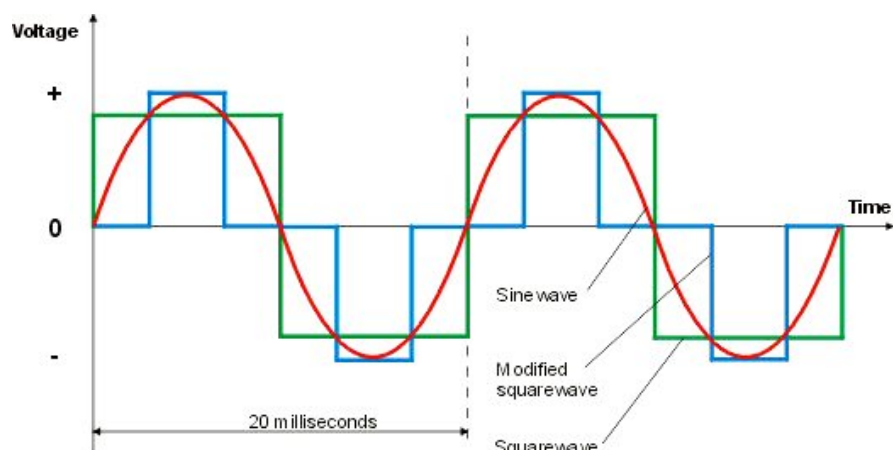


Figura 3. 27 - Onda quadrada e *modified sinewave*

A técnica de *modified sinewave* (analisada no ponto anterior para $\Phi < \pi$) pretende ser uma aproximação grosseira à função seno na frequência fundamental.

O SPWM é a técnica mais usada em inversores *true sinewave*, e que será implementada neste trabalho. Basicamente cada um dos comutadores do push-pull é comandado por um trem de impulsos cuja largura é proporcional ao valor da função seno num determinado ângulo de fase (Fig. 3.28 a) e b)).

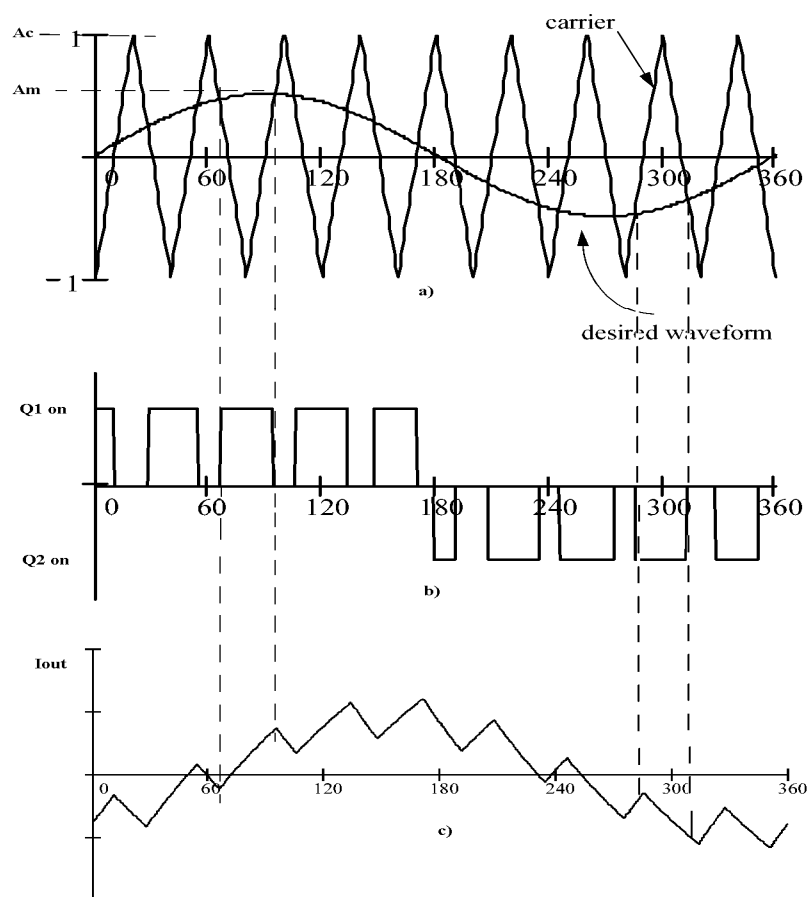


Figura 3. 28 - Modulação por SPWM

Este trem de impulsos é gerado comparando a função seno (sinal modulante) com uma onda triangular de frequência superior (portadora), como se pode ver na Fig. 3.28 a).

A Fig. 3.28 c) mostra um exemplo da onda de saída de corrente, depois de filtrada. Se se aumentar a frequência da portadora (e portanto a frequência de comutação) obtém-se melhores resultados para a onda de saída (Fig. 3.29).

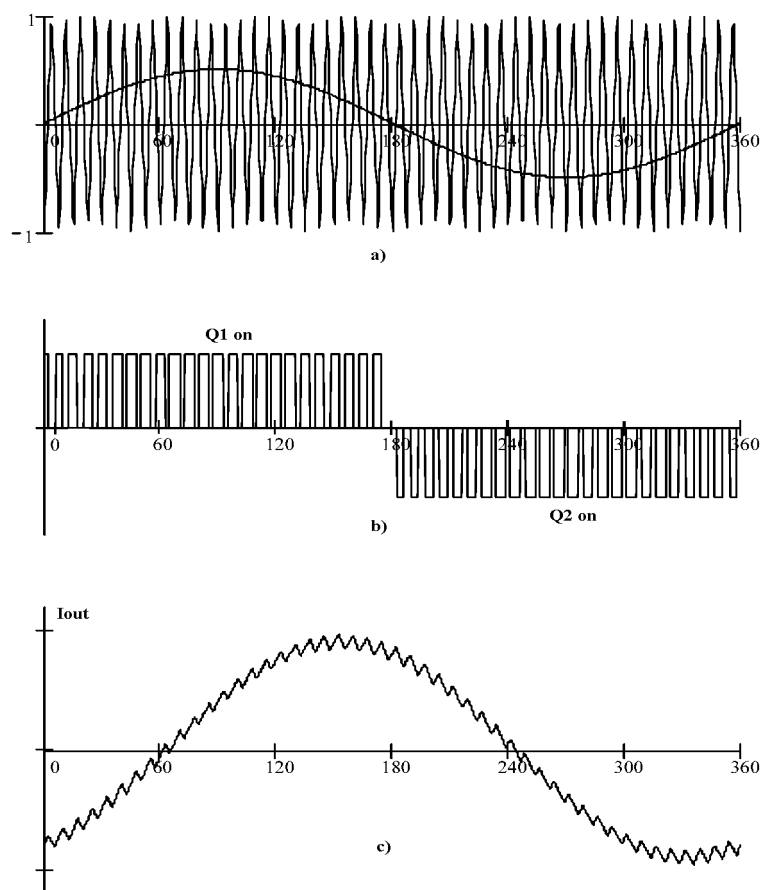


Figura 3. 29 - SPWM com maior frequência de comutação

Na Fig. 3.30 tem-se uma perspectiva do espectro que caracteriza o SPWM:

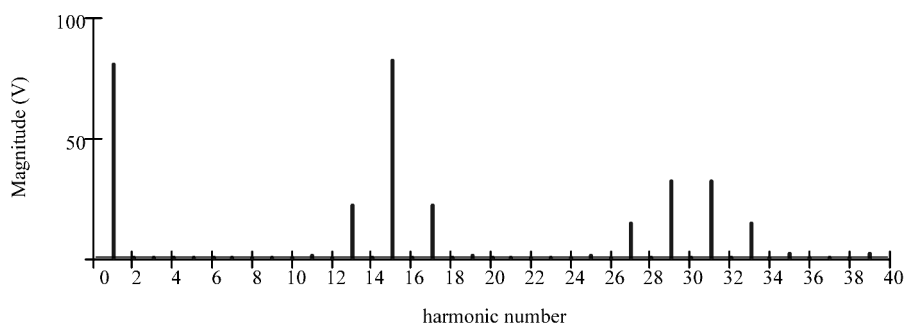


Figura 3. 30 - Espectro do SPWM para $n=15$ e $m=0,8$

Onde:

$$n = f_C/f_m \text{ e } m = A_m/A_C$$

f_C e A_C – Frequência e amplitude da portadora, respectivamente

f_m e A_m – Frequência e amplitude do sinal modulante, respectivamente

Verifica-se que existem duas componentes importantes para $n = 1$, que corresponde à frequência fundamental f_m , e para $n = 15$ que não é mais que a frequência da portadora f_C . Outros harmónicos aparecem para $n=13$, $n=15$, etc. Para o inversor deste trabalho apenas interessa a componente fundamental a 50hz. Todas as outras serão eliminadas por métodos de filtragem com o objectivo de minimizar a distorção harmónica total em relação à sinusóide pura.

3.4. Snubbers

Um dos grandes inconvenientes das fontes comutadas, onde se inclui o *push-pull* em particular, tem a ver com o facto dos elementos de comutação actuarem sobre cargas indutivas de uma maneira geral. Voltando ao exemplo do *push-pull* (Fig. 3.22), no momento em que Q1 fecha, uma corrente crescente começa a fluir no respectivo enrolamento do primário. O problema surge no instante em que Q1 abre porque essa corrente não se anula instantaneamente. Isto faz com que apareça aos terminais de Q1 um pico de tensão que pode ser muito elevado. Esta situação pode submeter o transistor a um elevado stress, pois este tem que lidar com condições de funcionamento próximas do seu limite máximo em termos de tensão e corrente. A Fig. 3.31 ilustra melhor o problema:

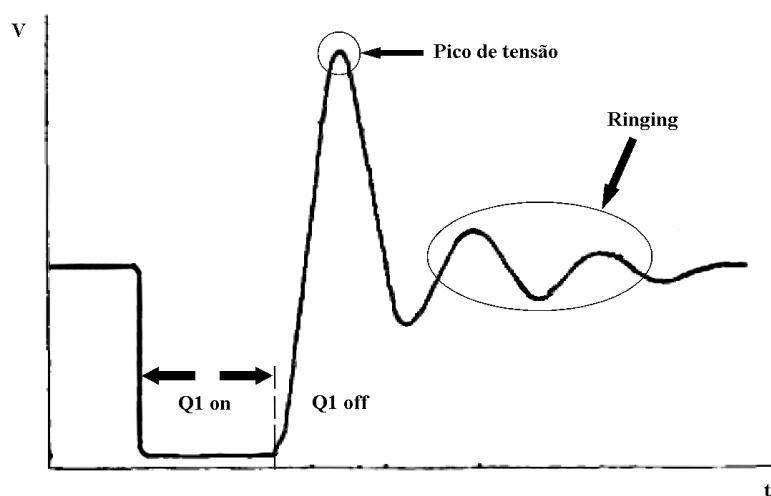


Figura 3. 31 - Forma de onda aos terminais de Q1

Este comportamento é devido à presença de indutâncias e capacidades parasitas. No caso do *push-pull* tem-se essencialmente a indutância de fugas do transformador e a sua capacidade distribuída que, no conjunto, modelam a forma de onda da Fig. 3.31.

É precisamente neste contexto que se insere o conceito de *snubber*. *Snubbers* não são mais que circuitos que são colocados aos terminais dos comutadores com os seguintes propósitos:

- reduzir ou eliminar *spikes* de corrente ou tensão;
- limitar o di/dt ou dv/dt ;
- transferir parte da potência dissipada no *switch* para uma resistência;
- reduzir as perdas totais associadas à comutação;
- reduzir as emissões electromagnéticas.

Encontram-se vulgarmente dois tipos de *snubbers*: o RC constituído por um condensador e uma resistência (Fig. 3.32) e o RCD, semelhante ao primeiro, onde se acrescenta um diódo (Fig. 3.33). O *snubber* RCD tem algumas vantagens sobre o RC, nomeadamente na diminuição das perdas no comutador e no próprio *snubber*. No entanto, para o mesmo valor de C_s , a tensão de pico é superior no RCD quando comparado com o RC.

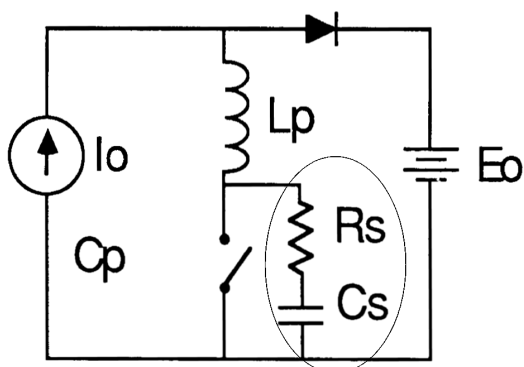


Figura 3. 32 - Snubber RC

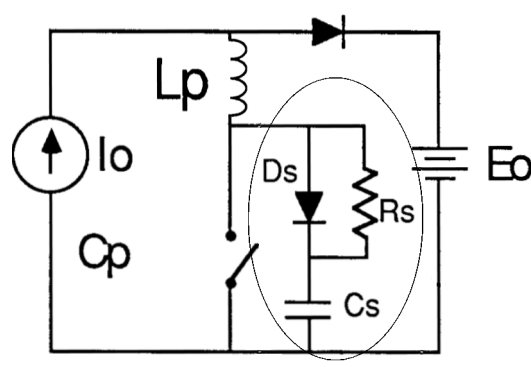


Figura 3. 33 - Snubber RCD

A Fig. 3.34 mostra a diferença de comportamento entre um *switch* com e sem *snubber*.

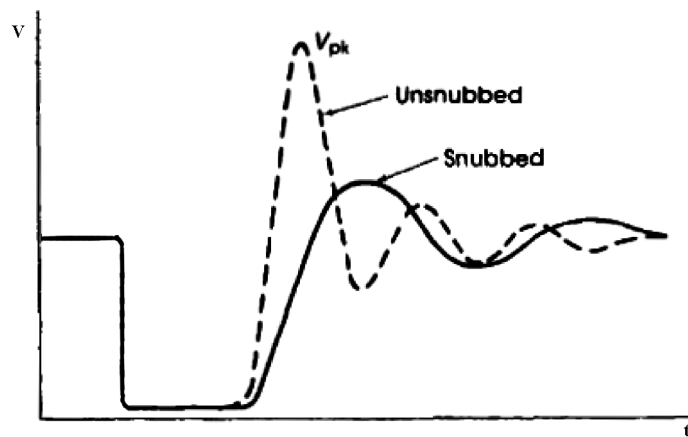


Figura 3. 34 - Q1 com e sem snubber

É importante realçar o facto dos *snubbers* dissiparem potência e portanto constituírem mais um factor de perdas. Tipicamente tem-se que:

$$P_{DISS} \approx C_S E_0^2 f_C \quad (3.23)$$

Onde:

P_{DISS} - Potencia dissipada no *snubber*

C_S - Valor da capacidade do *snubber*

f_C - Frequência de comutação

E_0 - Amplitude da fonte DC

A Eq. 3.23 é um resultado importante a ter em conta na hora de desenhar um *snubber*.

Existem várias técnicas mais ou menos empíricas para projectar *snubbers*. A maioria dessas técnicas assenta na análise inicial da forma de onda da Fig. 3.31 (em termos de tensão de pico e frequência de *ringing*) para estabelecer um ponto de partida para os valores de R e C. Esses valores são depois variados experimentalmente até se obter uma solução optimizada.

4. Modelo de Simulação

4.1. O Esquema e Seus Elementos

Para este trabalho foi construído um modelo de simulação em Pspice baseado na topologia *push-pull*. Pspice é um poderoso simulador digital genérico de circuitos analógicos que é usado para design, teste e verificação do funcionamento de um determinado circuito. Neste trabalho foi usado o OrCAD Pspice versão 15.7.

Pretende-se com este modelo observar o comportamento do inversor na situação em que é alimentado por um gerador DC de baixa tensão, ao mesmo tempo que se injecta energia na rede eléctrica. O nível de potência inserida deve ser ajustável e a forma de onda gerada no inversor tem de ser sinusoidal e deve estar em fase com a da rede. A Fig. 4.1 mostra o diagrama de blocos que resume o funcionamento do modelo usado.

Neste esquema a variável de controlo é a corrente de saída do inversor. Uma amostra dessa corrente é comparada com uma sinusóide de referência (*setpoint*) em fase com a rede. O sinal de erro resultante é depois processado por um compensador do tipo PID que, por sua vez, gera um sinal de controlo que comanda o gerador de PWM. Este último cria uma onda quadra de frequência programável, cujo *duty-cycle* é proporcional ao módulo da amplitude do sinal de entrada. Este sinal é depois aplicado a um gate driver de ataque aos transístores MOSFET M1 ou M2 (consoante o resultado do selector).

A Fig. 4.2 revela o circuito completo em Pspice usado para este modelo de simulação. Segue-se uma descrição um pouco mais detalhada de alguns dos elementos que compõem este circuito.

Diagrama de Blocos do Modelo Simulação do Inversor Push-Pull

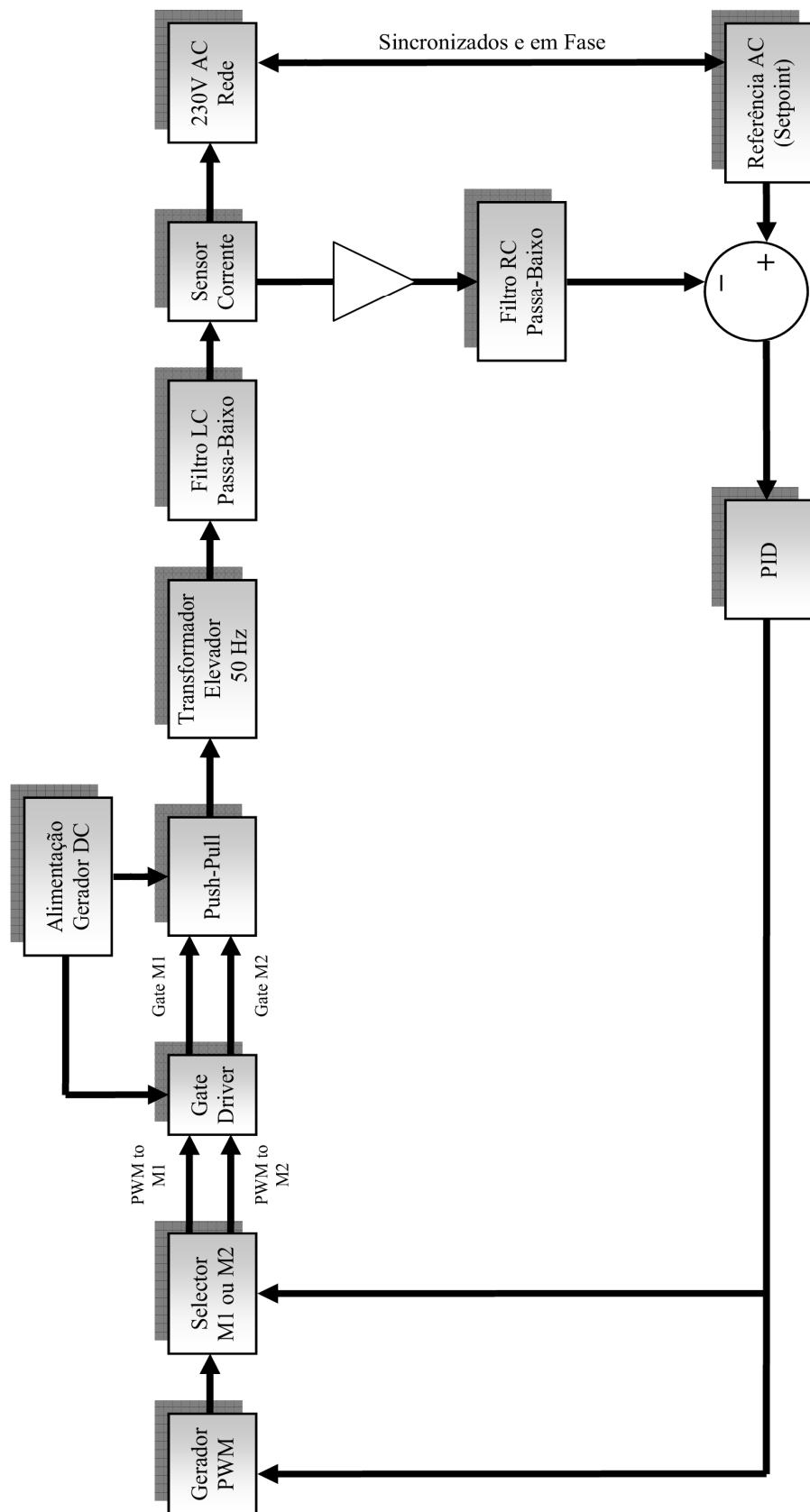


Figura 4. 1 – Diagrama de blocos do modelo de simulação

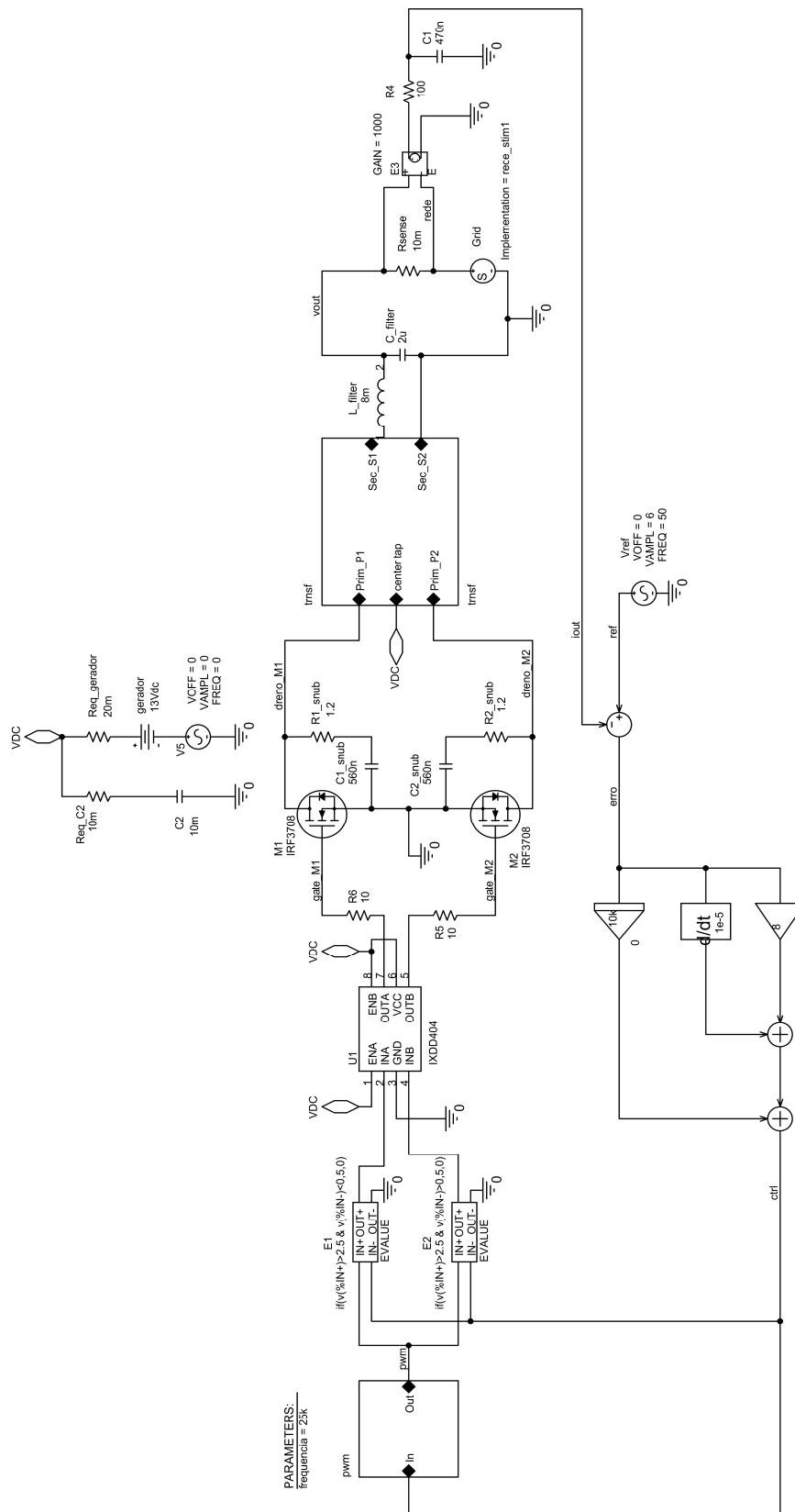


Figura 4. 2 - Circuito Pspice do modelo de simulação

- **Gerador de PWM**

PARAMETERS:
frecuencia = 25k

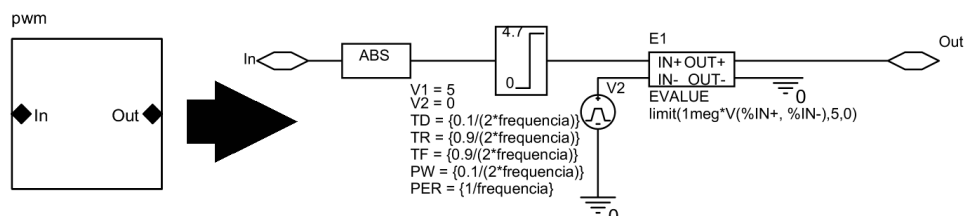


Figura 4.3 - Detalhe do gerador de PWM

Uma maneira de se poder simular alguns dos algoritmos que são executados num microcontrolador é usar blocos matemáticos em Pspice. É precisamente isso que se faz no esquema da Fig. 4.3. Neste bloco o sinal de entrada é convertido em módulo (só interessam valores positivos neste contexto) através do bloco ABS. De seguida o sinal é restringido ao intervalo [0V ; 4.7V] e depois comparado (por intermédio do módulo E1) com uma onda triangular de 0V a 5V (fonte V2). O intervalo de restrição usado para o sinal de entrada previne a sobremodulação, limitando o *duty-cycle* máximo da onda quadrada resultante a $4.7/5 = 94\%$. A frequência da onda triangular é um parâmetro programável.

- Transformador elevador a 50Hz

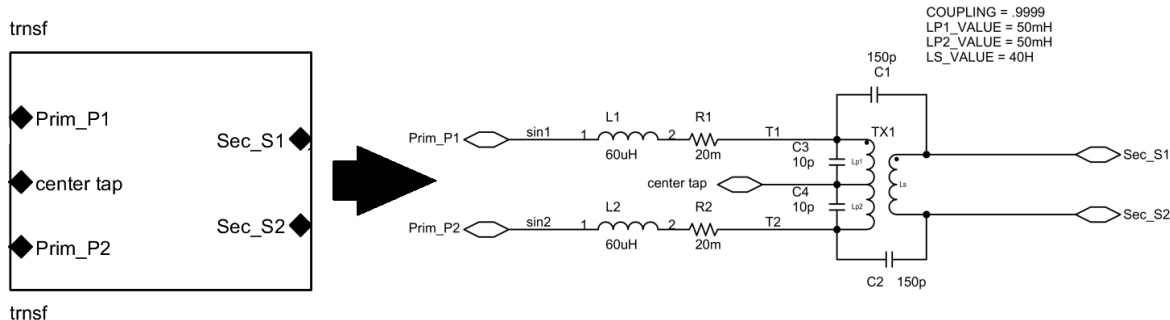


Figura 4.4 - Detalhe do transformador

O transformador elevador desta simulação foi feito com base num modelo de transformador ideal, onde se acrescentaram alguns elementos parasitas (Fig. 4.4). L1 e L2 representam as indutâncias de fugas do primário somadas à indutância de fugas do secundário reflectida no primário. R1 e R2 traduzem a resistência série equivalente do primário somadas à resistência série do secundário reflectida no primário. C1 a C4 correspondem à capacidade distribuída do transformador. Todos os valores atribuídos a estes componentes parasitas são plausíveis (para um transformador com núcleo de ferro)

e, portanto, não se basearam em nenhum transformador real em particular dado que esses valores não constam normalmente nos *datasheets* dos fabricantes. É importante referir que este modelo não considera as perdas no núcleo de ferro. A relação de transformação é ajustada manipulando os valores de indutância dos enrolamentos primários e secundário. Facilmente se demonstra que:

$$\frac{L_S}{L_P} = \left(\frac{N_S}{N_P} \right)^2 \quad (4.1)$$

Onde:

L_P e L_S - Valor da indutância no primário e secundário respectivamente

N_P e N_S - N° de espiras do primário e secundário respectivamente

- **Forma de onda da fonte Grid**

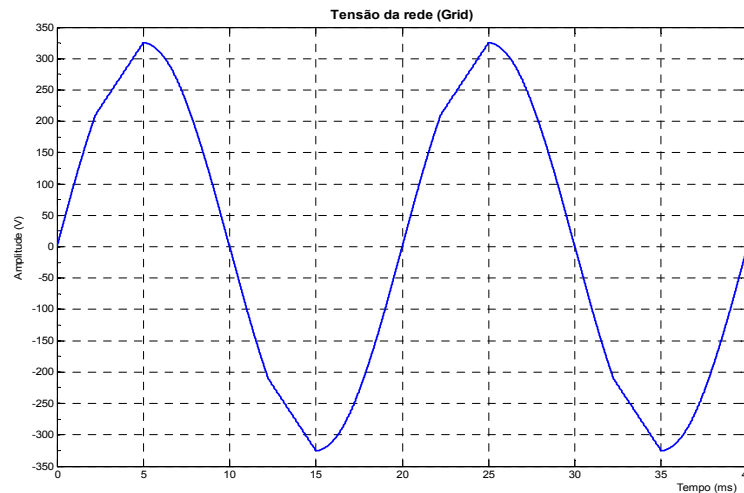


Figura 4. 5 - Forma de onda da rede usada neste modelo

Ao observar com um osciloscópio a forma de onda da rede pública verifica-se que esta não é propriamente uma sinusóide pura a 50Hz. Ela apresenta uma certa distorção especialmente nos flancos de subida antes da amplitude atingir o máximo. Esta distorção é provocada pelas fontes de alimentação dos dispositivos que estão ligados à rede.

Para reproduzir esta distorção de forma aproximada foi criada a forma de onda da Fig. 4.5. Basicamente usou-se uma sinusóide pura a 50Hz e fez-se um corte junto dos flancos de amplitude máxima. O ficheiro com os pontos gerados é depois executado pela fonte Vgrid, na Fig. 4.2. Com isto pretende-se avaliar o desempenho do inversor ligado a uma rede que possua alguma distorção desta natureza.

- **Selectores E1 e E2**

Os comparadores da Fig. 4.2 fazem parte da estratégia de controlo do inversor *push-pull* e funcionam do seguinte modo: se o sinal de saída do compensador PID for positivo então o sinal de PWM controla o transístor M2 (acção “*push*”). Se for negativo o sinal de PWM controla o transístor M1 (acção “*pull*”).

- **Gerador**

O bloco do gerador é composto por uma fonte DC (gerador) ideal em série com uma resistência (Req_gerador) e representa a resistência interna de um gerador real (ver Fig. 4.2). A tensão resultante é filtrada através de C2 em série com a sua resistência interna equivalente (Req_C2). A fonte de tensão AC V5 serve para adicionar uma variação na tensão de saída do gerador de modo a testar a capacidade de regulação do inversor nestas circunstâncias. Na realidade o que acontece é que o indivíduo ao pedalar não consegue manter um ritmo suficientemente estável para gerar uma tensão constante. É precisamente este factor que se pretende simular com a inclusão de V5 no modelo do gerador.

Relativamente aos restantes componentes, U1 representa o gate driver de ataque aos transístores M1 e M2, que é alimentado directamente a partir da tensão de saída do gerador (VDC). R5 e R6 são resistências de baixo valor que tipicamente se colocam à saída de uma gate driver para promover a estabilidade do seu funcionamento. E3 é um bloco matemático que mede a tensão diferencial aos terminais de Rsense (i.e. a corrente de saída do inversor) e depois amplifica o resultado por um factor de 10^3 .

4.2. Resultados da Simulação

Com base no circuito da Fig. 4.2, fez-se uma simulação no domínio do tempo, com um intervalo de 100ms. A tensão programada para o gerador é de 13V (sem perturbação, ou seja, V5 = 0V), a frequência do gerador de PWM de 25KHz e a amplitude de corrente pretendida é de 600mA. Como a amplitude da corrente de saída é dada por $V_{ref}/10$, programa-se Vref (ou seja o *setpoint*) com 6V de amplitude e uma frequência de 50Hz. No instante $t = 0ms$ tanto a fonte Grid como a fonte Vref começam com um ângulo de fase de 0° o que, sob ponto de vista da simulação, implica que ambas estão sincronizadas e em fase.

A Fig. 4.6 revela o comportamento dinâmico geral do inversor. A forma de onda da corrente de saída (Fig. 4.6 a)) é medida na resistência Rsense e o sentido de circulação é do inversor para a fonte Grid. Observa-se uma sinusóide a 50Hz (com algumas

componentes de alta-frequência, em que a dominante é a de 25KHz) com uma amplitude em torno de 600mA e em fase com a tensão da rede (Fig. 4.6 b)). Essa corrente equivale teoricamente a $(600\text{mA} \times 220\text{V})/1.41 = 97.6 \text{ W}$ de potência inserida na rede.

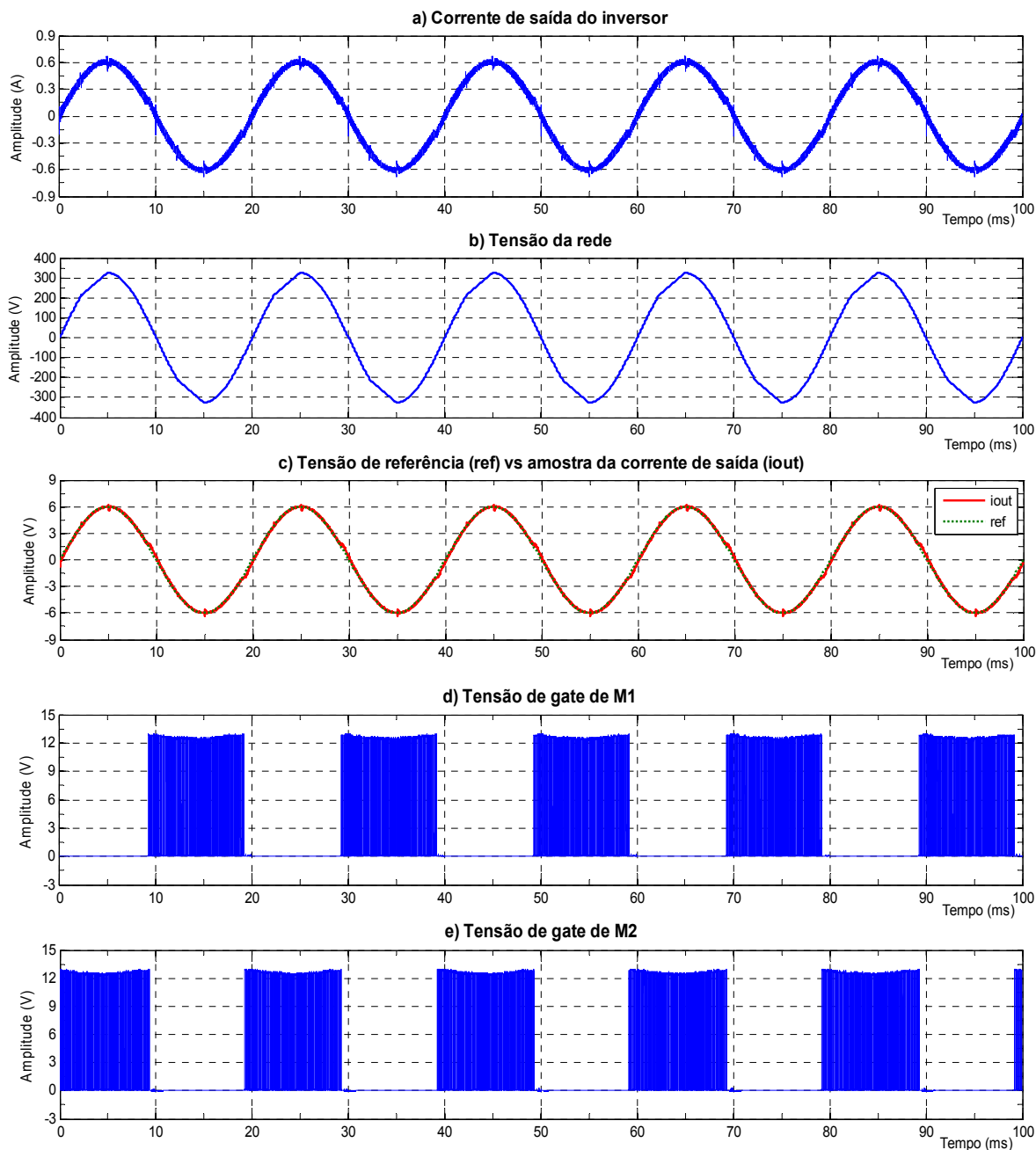


Figura 4. 6 - Comportamento dinâmico do modelo de simulação

O gráfico da Fig. 4.6 c) ilustra o bom desempenho do compensador PID em malha fechada, mantendo a amostra de corrente de saída (iout) muito próxima do valor de *setpoint* (ref). O ajuste do compensador é feito alterando o valor de ganho de cada um dos seus componentes. Numa primeira fase aumentou-se gradualmente o valor do ganho proporcional até um limite em que não se verificassem oscilações ou *overshoots* de

elevada amplitude. De seguida aumentou-se o valor do ganho do integrador até que o erro (diferença entre o *setpoint* e a amostra de corrente) fosse o mínimo possível, sem comprometer a velocidade de resposta do compensador. O componente derivativo acabou por não ser utilizado (o ganho foi programado com um valor bastante baixo). Na prática, o compensador utilizado é mais do tipo PI. Outro resultado importante do controlo deste circuito tem a ver com o facto da onda de corrente não possuir a deformação anteriormente descrita e que foi imposta para a tensão da rede (Grid). No limite um inversor com esta característica pode contribuir para melhorar o perfil sinusoidal da forma de onda da rede eléctrica.

A Fig. 4.6 d) e e) mostram o sinal de PWM que controla os transístores M1 e M2 respectivamente. Observa-se que o transístor M1 é responsável por modular as arcadas negativas da sinusóide de corrente enquanto que M2 modula as arcadas positivas. Nota-se uma pequena diminuição da amplitude do sinal de PWM nas zonas onde a amplitude da corrente de saída é máxima. Isto acontece porque o sinal tem origem nas saídas do gate driver U1 que é alimentado via VDC. Quanto mais corrente se retira do gerador, maior será a queda de tensão na sua resistência interna ($R_{eq_gerador}$) e, portanto, o valor de VDC diminui (isto é claramente observável no gráfico da Fig. 4.8 e)). Na Fig. 4.7 é possível ver com um pouco mais de detalhe a modulação em SPWM e o modo como varia a largura dos pulsos.

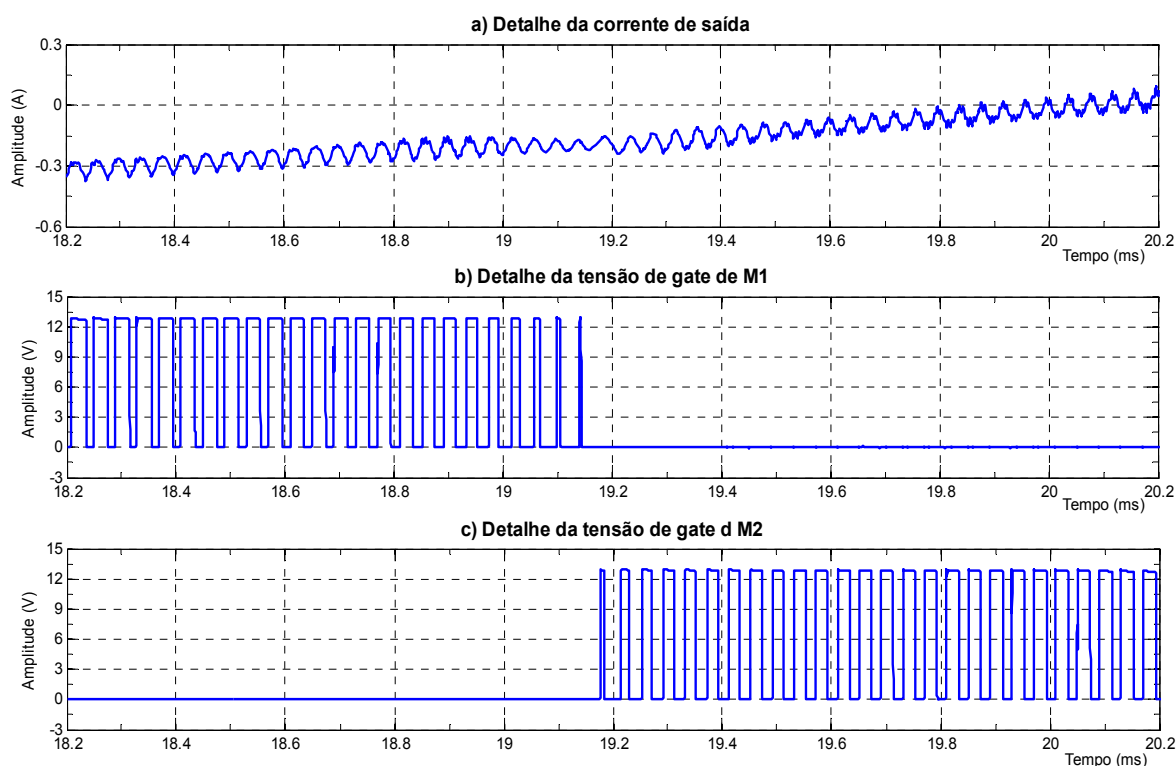


Figura 4. 7 - Pormenor da modulação por SPWM

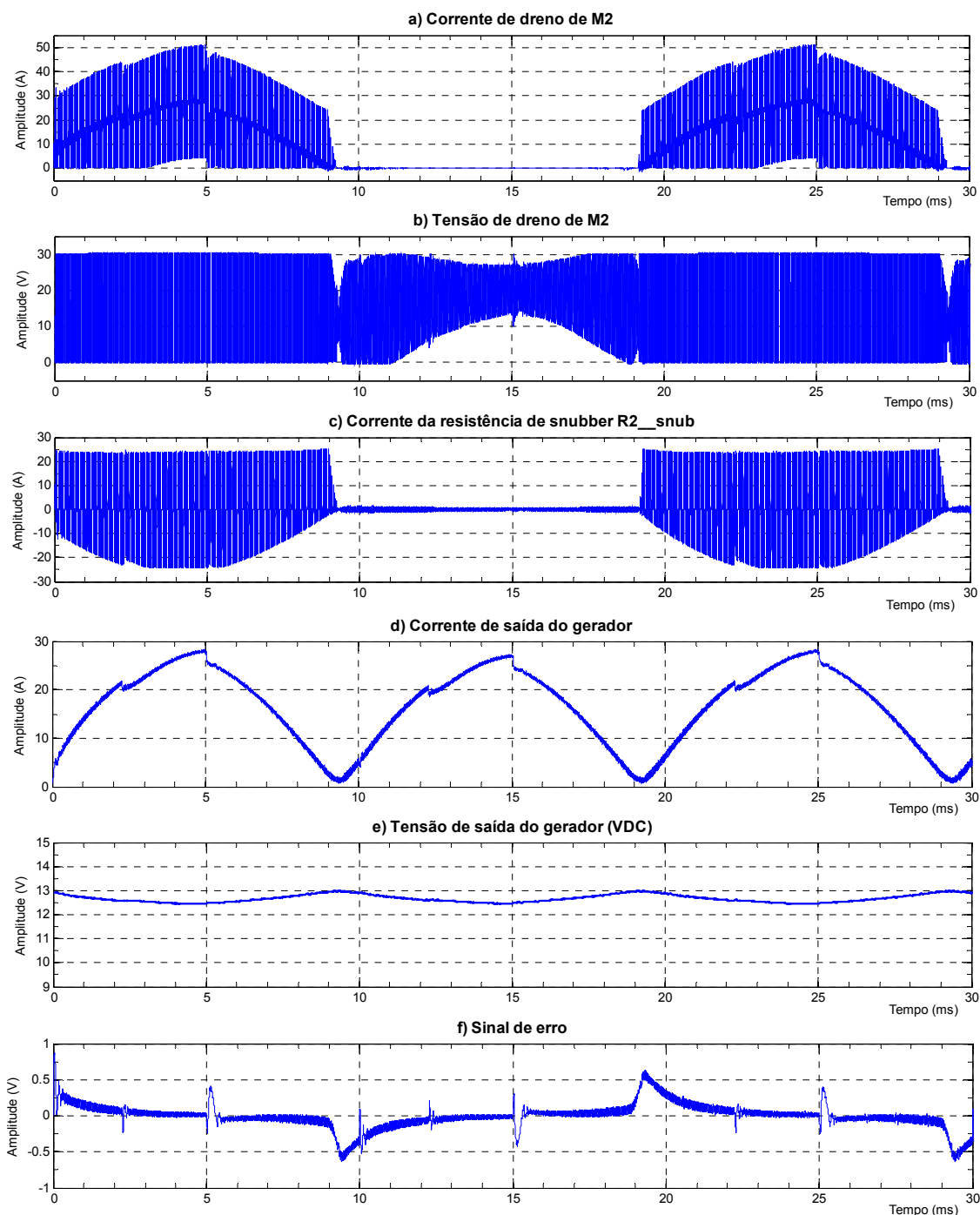


Figura 4. 8 - Formas de onda de outros pontos do circuito

A Fig. 4.8 ilustra as formas de onda mais relevantes de outros pontos do circuito. Destaque para a corrente de saída do gerador (gráfico d) da mesma figura) em que os máximos coincidem exactamente com os instantes onde a amplitude da onda de saída do inversor é máxima (isto porque se considerou que a rede é vista como uma carga puramente resistiva, isto é, $\cos\phi = 1$). No gráfico f) é possível observar a evolução do sinal de erro. Apresenta-se muito próximo do valor 0 na maior parte do tempo, contudo agrava-se próximo das passagens por zero (10ms, 20ms, etc). Isto acontece porque é

nesses instantes que ocorre a comutação entre M1 e M2, isto é, M1 deixa de modular por PWM e passa a funcionar M2 ou vice-versa. Nesta sequência, a malha de controlo acaba por fazer convergir a variável de controlo com o *setpoint* e o sinal de erro volta a decair para valores em torno de zero o que garante a estabilidade de todo o sistema.

A potência média fornecida pelo gerador é de 207,1W. Este resultado foi obtido medindo o valor médio do produto entre a corrente que atravessa Req_gerador e VDC. O programa de simulação permite calcular estes parâmetros de forma directa. Os transístores M1 e M2 juntos dissipam, em média, uma potência de 87.5W, o que equivale a 42.2% da potência entregue pelo gerador. Apesar de serem transístores com baixa resistência dreno-fonte (no estado on cerca de 12mΩ), o que acontece é que os transitórios (que acontecem nas comutações para off) provocam picos de tensão (rever Fig. 3.31 do capítulo 3.4) que rapidamente atingem o limite de disrupção dos transístores (cerca de 30V com base na Fig. 4.8 b)). A tensão de disrupção (*breakdown voltage*) dreno-fonte de um transistor MOSFET é o valor acima do qual a junção pn entre o dreno e o substracto (que está inversamente polarizado) sofre uma ruptura de avalanche (*weak avalanche breakdown*) que se traduz num aumento súbito da corrente de dreno, apesar do transistor estar ao corte ($V_{gs} = 0V$). Nesta situação M1 e M2 acabam por conduzir alguma corrente o que contribui para a dissipação de potência referida. Os *snubbers* constituídos por R1_snub, C1_snub, R2_snub e C2_snub atenuam parte desse fenómeno, absorvendo alguma da energia contida nesses picos de tensão, mas não o eliminam totalmente. A fonte Grid, que representa a rede eléctrica, acabou por absorver cerca de 94,7W o que resulta numa eficiência de 45.7%. A restante potência fornecida pelo gerador (cerca de 12.1% ou 25.1W dos 207.1W) é dissipada nos demais elementos do circuito.

Testou-se a eficiência do modelo de simulação em função de alguns valores de tensão do gerador. As restantes condições desta simulação são as mesmas da simulação anterior.

Tensão no gerador	Eficiência
13V	45.7%
15V	39.3%
17V	34.4%
20V	28.9%

Tabela 4. 1 – Eficiência em função da tensão do gerador

A tabela 4.1 mostra os resultados obtidos. Verifica-se que quanto mais baixa for a tensão do gerador melhor será o rendimento deste modelo. No entanto, para tensões abaixo de 12V, constata-se que a distorção da onda de corrente de saída começa a ser demasiado elevada e instável.

Um outro teste no domínio do tempo foi realizado com o objectivo de avaliar a capacidade de regulação deste inversor na situação em que a tensão do gerador flutua dentro de uma determinada gama de valores. Para isso fixou-se o valor da fonte DC gerador nos 13V e programou-se a fonte de perturbação V5 com uma amplitude de 2V, uma frequência de 80Hz e um offset nulo. A frequência do gerador de PWM é de 25KHz e a amplitude de corrente de saída pretendida é de 600mA ($V_{ref} = 6V$).

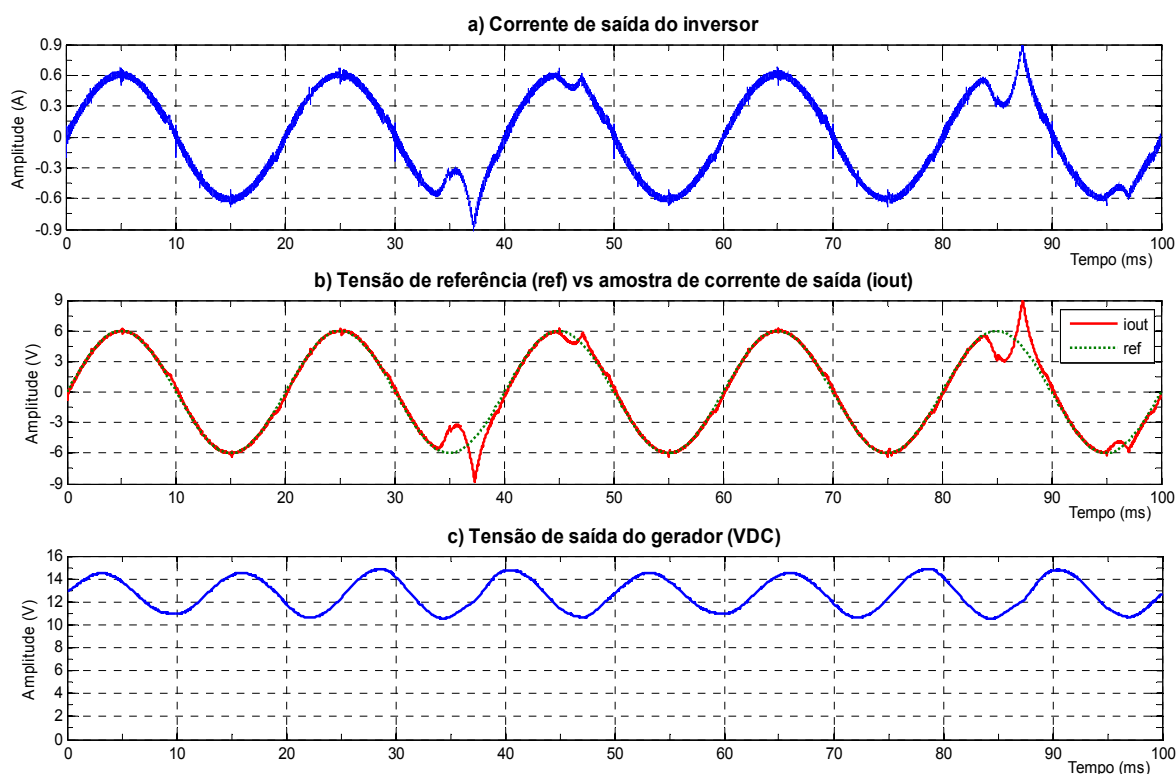


Figura 4. 9 - Resultado da simulação com perturbação da tensão de entrada

A Fig. 4.9 mostra os resultados deste teste. Apesar da tensão de entrada do inversor ter uma variação considerável (4V de pico-a-pico), o modelo apresenta uma boa capacidade de regulação, mantendo a corrente de saída dentro dos parâmetros pretendidos (a nível de amplitude, frequência e ruído). No entanto notam-se pequenas divergências nos instantes $t = 35ms$, $t = 45ms$, $t = 85ms$ e $t = 95ms$. Estas distorções coincidem com os instantes onde a tensão do gerador é mais baixa. Esta característica impõe um nível de tensão mínimo para o gerador de modo a garantir o correcto funcionamento do inversor.

5. Construção do Protótipo

5.1. Rolo de Treino com Gerador DC

Para este trabalho foi construído um protótipo de um aparelho de exercício do tipo cardiovascular baseado num rolo de treino. Um rolo de treino é um dispositivo que permite acoplar uma bicicleta vulgar e transformá-la num aparelho de exercício estático. Essencialmente é composto por um par de braços reguláveis onde se encaixa o eixo traseiro da bicicleta e que estão montados numa base robusta e suficientemente estável para manter o conjunto na vertical (incluindo o atleta). A roda traseira assenta sobre um rolo livre que absorve toda a energia cinética produzida no acto de pedalar. O protótipo deste trabalho, para além de possuir as características anteriormente mencionadas, inclui também um gerador DC que está engrenado ao rolo livre. Foi aproveitado de um outro dispositivo um pequeno motor DC de 24V/80W e montado a funcionar como gerador. Toda a estrutura foi feita a partir de chapa e tubo de ferro.



Figura 5. 1 – Protótipo do rolo de treino já com uma bicicleta montada

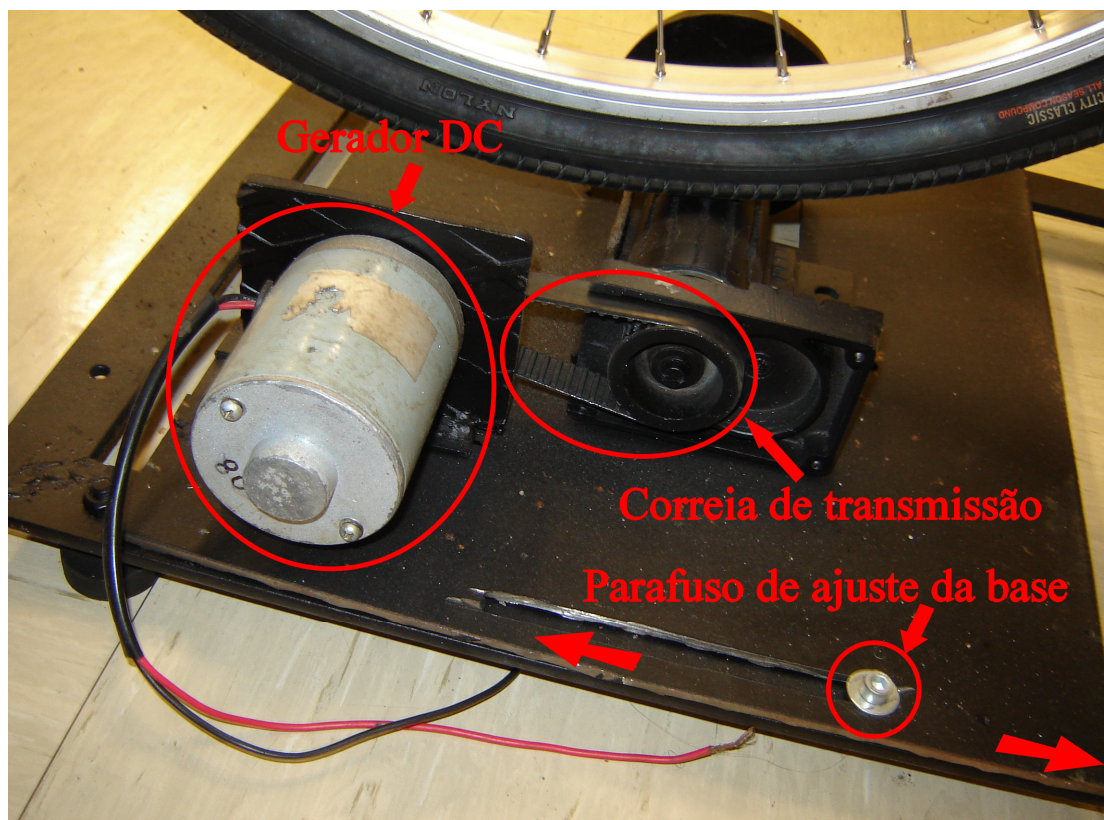


Figura 5. 2 - Detalhe do gerador DC e demais elementos



Figura 5. 3 - Detalhe dos braços de apoio e veios de fixação



Figura 5. 4 - Pormenor de uma das quatro bases de nivelamento

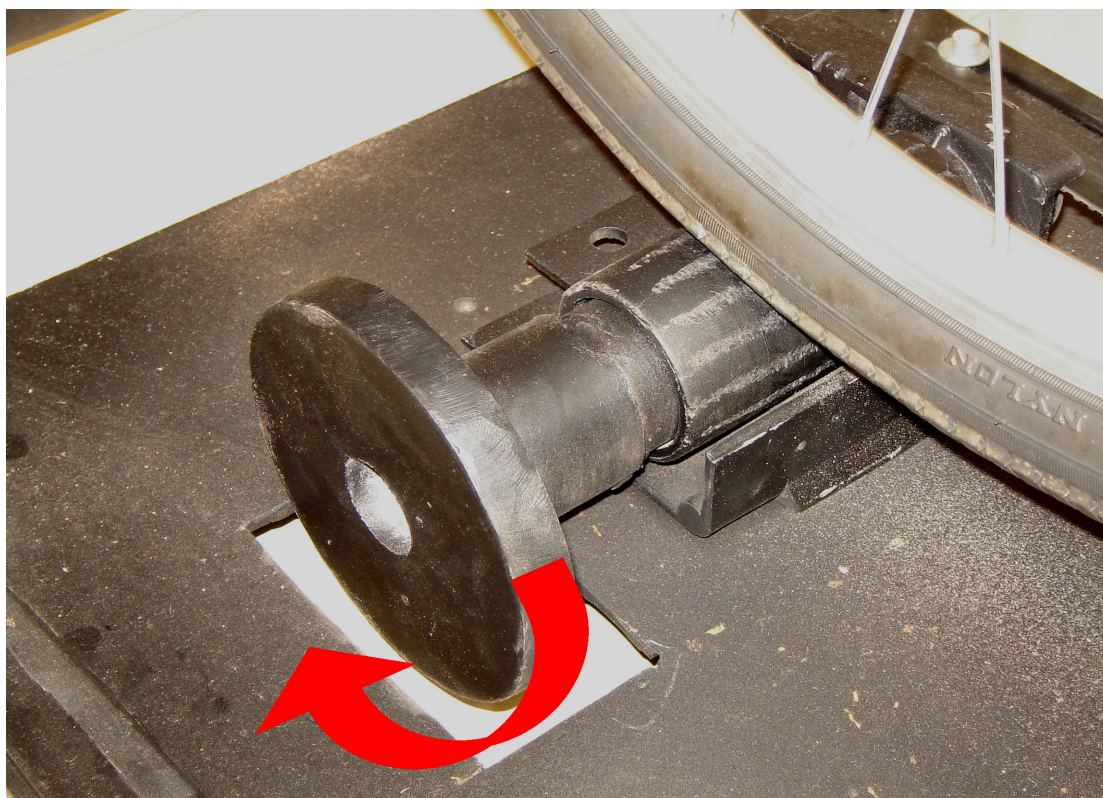


Figura 5. 5 - Roda de inércia do rolo de treino

Na Fig. 5.1 encontra-se uma foto com o aspecto final do rolo de treino já com uma bicicleta incorporada. Na Fig. 5.2 é possível observar com um pouco mais de detalhe o pequeno gerador DC engrenado ao rolo através de uma correia. Antes deste protótipo, houve um primeiro em que a ligação entre o gerador e o rolo era feita através de uma corrente. Contudo essa solução veio-se a revelar pouco eficaz uma vez que a corrente estava sempre a “saltar” ou a encravar. O problema residia na roda dentada que estava aplicada no eixo do gerador pois era demasiado pequena (e concebida manualmente, o que implica muito menor precisão no seu dimensionamento). Daí que se tenha optado por uma correia dentada. A placa metálica onde assentam o rolo e gerador pode ser ajustada longitudinalmente (na direcção do quadro da bicicleta) através do ajuste ilustrado na Fig. 5.2. A imagem da Fig. 5.3 mostra o pormenor do sistema de encaixe do eixo traseiro da bicicleta nos braços de apoio do aparelho de treino. É possível regular o afastamento dos veios de fixação assim como o ângulo dos braços de apoio em relação à base (estes rodam quase 180° em torno de um eixo) o que permite encaixar no aparelho praticamente qualquer tamanho ou tipo de bicicleta. É também possível ajustar o nivelamento do aparelho (Fig. 5.4) de modo a garantir maior estabilidade. O eixo do rolo inclui uma roda de inércia (Fig. 5.5) que serve para evitar que a rotação do rolo varie muito rapidamente, o que contribui para o conforto de quem pedala com este aparelho (no fundo a roda de inércia não é mais do que um filtro passa-baixo mecânico). Esta roda foi construída e aplicada manualmente pelo que não ficou centrada com muita precisão. Em consequência disso o rolo passou a vibrar demasiado quando atingia determinada velocidade de rotação. Uma forma de minimizar este problema foi soldar estrategicamente pequenos pedaços de ferro (que actuam como contrapesos) na face interior da roda de inércia, compensando esse desequilíbrio. O factor de desmultiplicação da roda traseira para o gerador é de 1 para 22, i.e., para cada volta completa que a roda faz o gerador faz 22 voltas.

Testou-se o protótipo aplicando uma carga aos terminais do gerador (uma pequena lâmpada de halógeno de 24V/150W) e obteve-se uma tensão média de 16V e uma corrente média de 3A o que perfaz uma potência de $16 \times 3 = 48\text{W}$. O nível de esforço físico para obter este resultado foi elevado e de curta duração. Para níveis mais moderados de esforço físico a potência disponível será menor que 48W. As perdas mecânicas do aparelho são significativas e contribuem de forma decisiva para o resultado obtido.

5.2. Circuito Inversor

Neste ponto do trabalho é apresentado em pormenor o projecto e construção do circuito inversor baseado na topologia *push-pull*. A estrutura base deste circuito assenta no modelo de simulação implementado onde se acrescentaram outras funcionalidades importantes e que serão descritas neste capítulo. Segue-se a apresentação dos esquemas e respectiva discussão dos vários elementos que compõem este circuito.

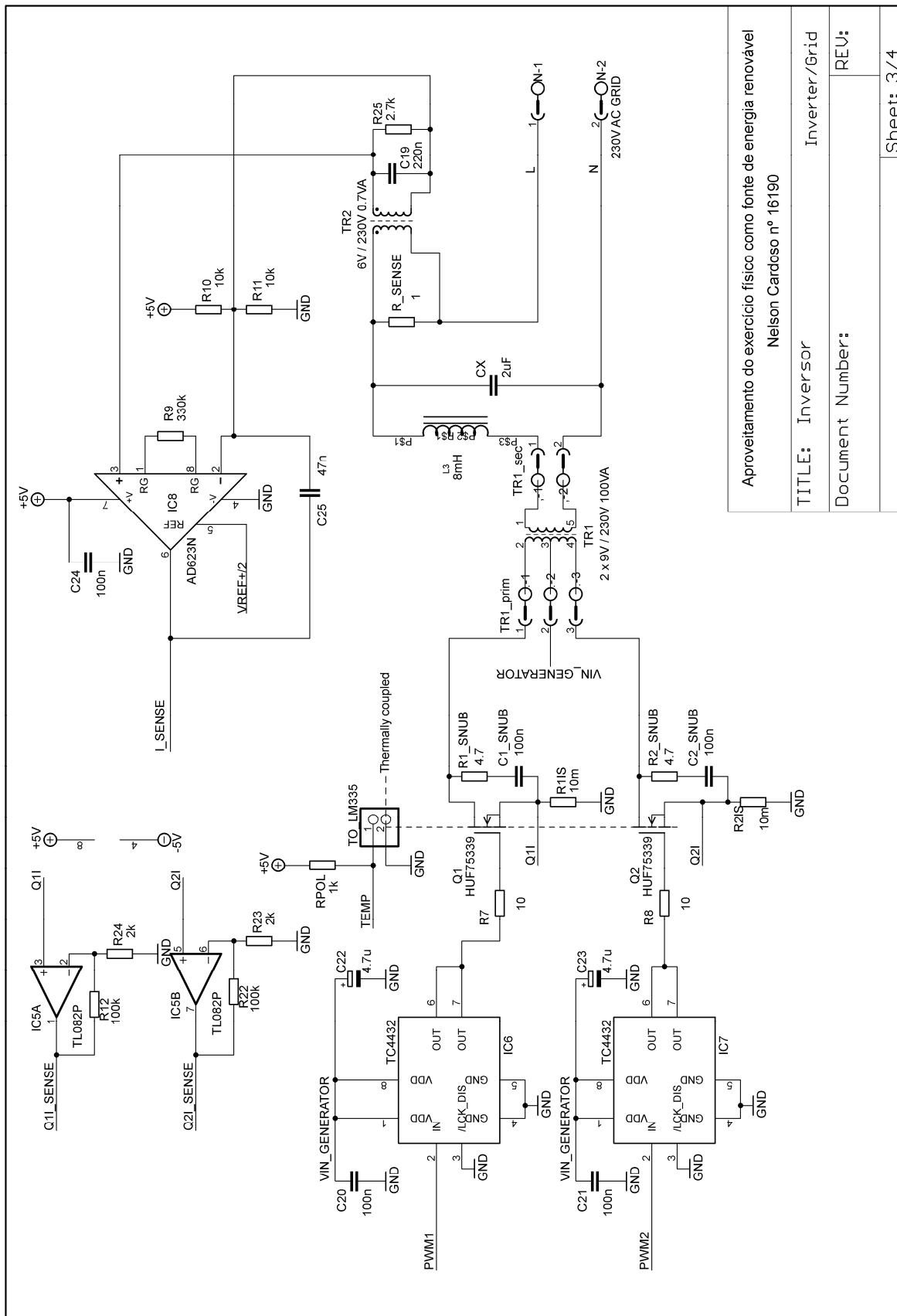


Figura 5. 8 - Esquema do inversor

Aproveitamento do exercício físico como fonte de energia renovável
Nelson Cardoso nº 16190

TITLE: Inversor Inverter/Grid

Document Number: REV:

Sheet: 3/4

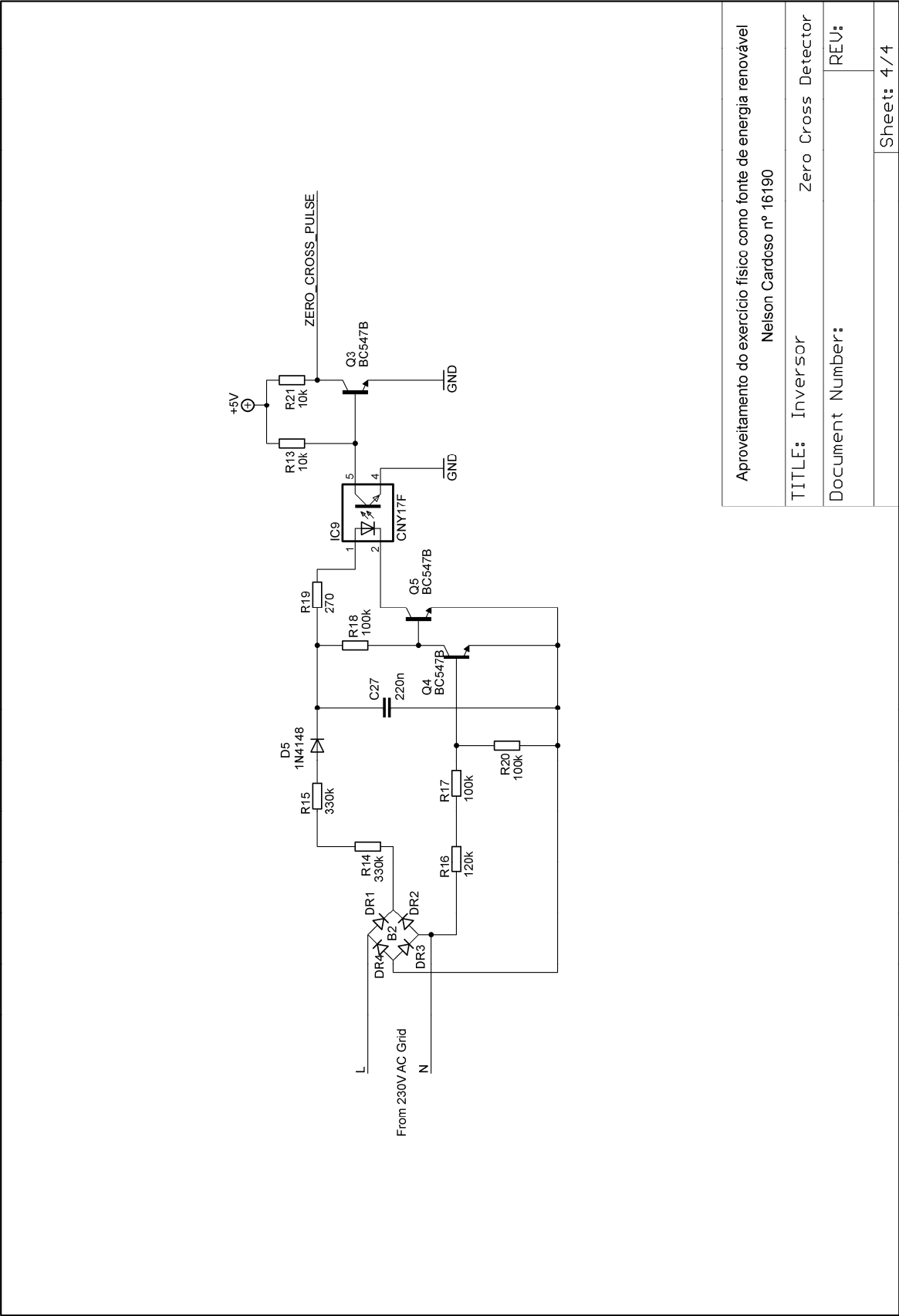


Figura 5. 9 - Esquema do detector de passagem por zero

Aproveitamento do exercício físico como fonte de energia renovável		
Nelson Cardoso nº 16190		
TITLE: Inversor	Zero Cross Detector	
Document Number:	REV:	
		Sheet: 4/4

Conforme se pode ver na Fig. 5.6, o circuito funciona em torno do microcontrolador de baixo custo dsPIC4011 da Microchip. Trata-se de um controlador digital de sinal de 16 bits e suporta até 30 MIPS com uma frequência máxima de 120MHz. Funciona dentro de uma gama de tensões que vai dos 2.5V até aos 5.5V, com baixo consumo de energia. Possui memória do tipo flash para programa com 48KB, 2KB de RAM e uma EEPROM não volátil de 1KB. Não é o objectivo deste capítulo fazer um levantamento exaustivo de todas as potencialidades deste microcontrolador. Contudo, vale a pena destacar algumas das suas características mais importantes para este trabalho, nomeadamente ao nível dos periféricos:

- Portos de I/O de elevada corrente (*sink/source* de 25mA)
- 5 *timers* de 16 bits e 2 *timers* de 32 bits
- 3 módulos de PWM num total de 6 canais disponíveis
- 9 entradas analógicas de 10 bits que partilham um conversor A/D de 4 canais do tipo *sample and hold*
- 2 módulos de comunicação UART e um módulo SPI

Possui também três modos de funcionamento de baixo consumo energético (*idle*, *sleep* e *low-power RC oscillator*). Uma outra funcionalidade extremamente útil é a possibilidade de poder ser programado na própria placa de circuito da aplicação a que se destina, o que resulta numa maior flexibilidade e poupança de tempo de desenvolvimento. Em suma, trata-se de um dispositivo cujas características são adequadas para a construção deste inversor.

Voltando ao esquema da Fig. 5.6, a tensão positiva de referência para a ADC do dsPIC é garantida através de IC2 com um valor de 4.096V e precisão de 1% a 25° C. A tensão de referência foi escolhida de modo a que $4.096/2^{10} = 4\text{mV}$ de resolução numa escala de 10 bits, o que facilita posteriormente os cálculos de conversão dos valores da ADC. A interface de comunicação RS232 é feita através de IC3 (MAX232). O conector JP1 faz a ligação entre IC1 e o programador externo PICKit 2 da Microchip enquanto que JP2 permite ligar outros dispositivos via SPI. JP3 é um porto de expansão que disponibiliza os pinos livres de IC1 para permitir a possibilidade de acrescentar outras funcionalidades futuras a este inversor, como por exemplo um pequeno *display* LCD e botões de comando.

O esquema da Fig. 5.7 representa a fonte de alimentação de 5V que alimenta a parte da electrónica responsável pelo acondicionamento de sinal e controlo. A tensão é gerada a partir de IC4 que não é mais do que um regulador comutado de 5V baseado na topologia *buck* (também conhecida por *step-down*) através da malha formada por D2, L1 e C14. Os componentes circundantes foram escolhidos segundo as indicações no respectivo *datasheet*. Este regulador suporta tensões de entrada desde 8V até 40V e é capaz de fornecer uma corrente máxima de 3A. Para estas condições, o regulador IC4 é bem mais eficiente do que os reguladores lineares (por exemplo o clássico LM7805) razão pela qual foi escolhido para este projecto. Este regulador pode ser alimentado a

partir do próprio gerador ou a partir de uma fonte externa. D6 e D7 garantem que não ocorrem correntes inversas. Cx1 e Cx2 são condensadores electrolíticos de filtragem para a tensão de entrada vinda do gerador enquanto que R5 e R6 constituem um divisor resistivo que reduz o valor dessa tensão de forma a poder ser monitorizada pelo dsPIC.

Na Fig. 5.8 encontra-se o esquema do inversor na topologia *push-pull*. O elemento central deste circuito é o transformador toroidal com núcleo de ferro (TR1). Do lado do primário tem-se dois enrolamentos para 9V, no secundário um enrolamento para 230V (valores eficazes) e uma potência aparente de 100VA. Os comutadores são constituídos por dois transístores MOSFET de potência Q1 e Q2. Estes transístores da Fairchild possuem uma tensão de disrupção dreno-fonte de 55V, uma resistência dreno-fonte de $12\text{m}\Omega$ (para $V_{GS} = 10\text{V}$) e são capazes de lidar com correntes até 75A (em modo contínuo). As correntes que atravessam Q1 e Q2 são detectadas através das resistências R1IS e R2IS. Esses sinais são depois amplificados pelos *opamp's* IC5A e IC5B, na configuração não inversora, e entregues a um par de entradas analógicas de IC1. A alimentação negativa é retirada do MAX232 e filtrada por L2 e C26 (Fig. 5.6). O ganho de, por exemplo, IC5A é dado por $1+R12/R24 = 51$. A temperatura dos transístores é detectada por um sensor LM335. Este sensor tem uma sensibilidade de $10\text{mV}/^\circ\text{K}$ e é polarizado por intermédio de RPOL. A tensão resultante é monitorizada pelo microcontrolador. IC6 e IC7 são os *gate-drivers* que recebem os sinais de controlo vindos dos módulos de PWM (pulsos com uma amplitude máxima de 5V) de IC1 e os transformam em pulsos de ataque aos MOSFET's, com uma amplitude máxima 0.8V abaixo da tensão do gerador. Estes integrados funcionam com uma tensão de alimentação que vai de 4.5V até 30V (fornecida directamente pelo gerador) e são capazes de debitar à saída uma corrente de pico de 1.5A suficiente para carregar uma capacidade de 1nF em 25nseg. Os seus tempos de atraso são inferiores a 78nseg, mais do que suficiente para trabalharem a frequências em torno de dezenas de KHz. No secundário de TR1 a tensão de saída é filtrada por um filtro passa-baixo LC (filtro de 2ª ordem) composto por L3 e CX e cuja frequência de corte é de 1258Hz (a frequência de corte é dada por $1/(2\pi(LC)^{0.5})$). A corrente de saída (variável de controlo) é medida pela queda de tensão na resistência R_SENSE. O sinal obtido é amplificado pelo pequeno transformador TR2 que garante o necessário isolamento galvânico. O ganho de sinal obtido com TR2 é demasiado elevado para a ADC de IC1 (a tensão máxima permitida é de 4.096V, i.e. VREF+), daí que se tenha ligado no secundário uma resistência de carga, R25, de modo a diminuir esse valor. O condensador C19, para além de filtrar o sinal de corrente, o seu valor foi determinado experimentalmente de modo a compensar o desvio de fase provocado pelo comportamento indutivo de TR2. Ou seja, o sinal sinusoidal sofre um pequeno desvio de fase quando passa por TR2 e é compensado com a inclusão de C19.

Numa primeira abordagem, a corrente de saída do inversor era detectada usando um circuito integrado de baixo custo baseado no efeito de Hall (o ACS714 da Allegro para 5A). Este integrado garantia isolamento galvânico, usava apenas uma alimentação de 5V

e tinha um comportamento linear dentro da gama 0A a 5A. À primeira vista parecia a solução ideal para este inversor. Contudo a sua sensibilidade era apenas de 185mV/A e isso reflectiu-se nos primeiros ensaios experimentais deste inversor. Para baixas potências, por exemplo 50W, a amplitude máxima da corrente de saída é de $(50 \times 1.41)/230 = 0.31\text{A}$ o que dá $185 \times 0.31 = 57\text{mV}$ de amplitude máxima à saída do sensor. Nestas condições, verificava-se que a relação sinal/ruído era extremamente pobre para poder ser usado como feedback para a estratégia de controlo. De facto a relação sinal/ruído deste sensor melhora significativamente para correntes superiores e que vão para além das especificações deste protótipo. Esta solução acabou por ser descartada e não foi possível encontrar no mercado outro sensor económico, com características semelhantes e de maior sensibilidade. Optou-se então pela solução descrita no parágrafo anterior, isto é, recorrendo a um pequeno transformador.

Continuando com o trajecto do sinal de corrente na Fig. 5.8, IC8 é um amplificador de instrumentação com entradas diferenciais onde são ligadas as saídas de TR2. O divisor resistivo, formado por R10 e R11 na entrada negativa de IC8, acrescenta um offset de referência de 2.5V ao sinal de corrente de modo a garantir que essa sinusóide oscile dentro de uma gama de valores positivos, uma vez que IC8 é alimentado apenas com +5V. Uma das principais razões que levou à escolha deste amplificador para acondicionar o sinal é o facto da saída poder ser referenciada a $V_{REF}/2$ o que permite aproveitar ao máximo toda a gama da ADC (valores entre 0V e 4.096V). O condensador C25 providencia uma filtragem adicional.

O esquema da Fig. 5.9 representa o detector de passagem por zero com isolamento galvânico. Este circuito gera um pulso de 5V cada vez que a tensão da rede cruza a referência (0V) apenas nas transições positivas da sinusóide, i.e. nas passagens da zona negativa para positiva. Por cada período há duas passagens por zero (que correspondem às fases 0° e 180°), mas por uma questão de sincronismo, apenas interessa uma delas. Desse modo garante-se que as passagens por zero são detectadas sempre na mesma fase o que simplifica imenso o processo de sincronização deste inversor. Os pulsos são posteriormente aplicados ao pino 17 de IC1 que gera uma interrupção cada vez que transita do estado lógico 0 para 1. O funcionamento do circuito é bastante simples. Durante a arcada positiva da tensão da rede (tensão no nó L > N) o transistor Q4 está cortado (porque DR3 está directamente polarizado) e Q5 está em condução. A corrente que passa em R19 é suficientemente baixa para não activar o optoacoplador (IC9) e portanto Q3 permanece em condução. Nesta fase a tensão em C27 é muito baixa (cerca de 1.25V). Quando ocorre a transição para a arcada negativa (tensão no nó L < N), Q5 corta e Q4 entra em condução (porque DR3 está inversamente polarizado). A corrente passa agora por R18 e a tensão em C27 começa a subir. Nesta fase Q3 permanece em condução. Assim que se dá a transição novamente para a arcada positiva, Q4 corta e Q5 entra em condução, só que agora a energia armazenada em C27 é suficiente para activar momentaneamente o optoacoplador via R19. Nesta fase Q3 corta e no seu colector aparece uma tensão de 5V. Assim que C27 descarregar, IC9 desactiva e Q3 volta a

entrar em condução. O díodo D5 garante que o condensador C27 apenas descarrega na direcção de R19. A duração do pulso gerado depende da quantidade de carga armazenada em C27.

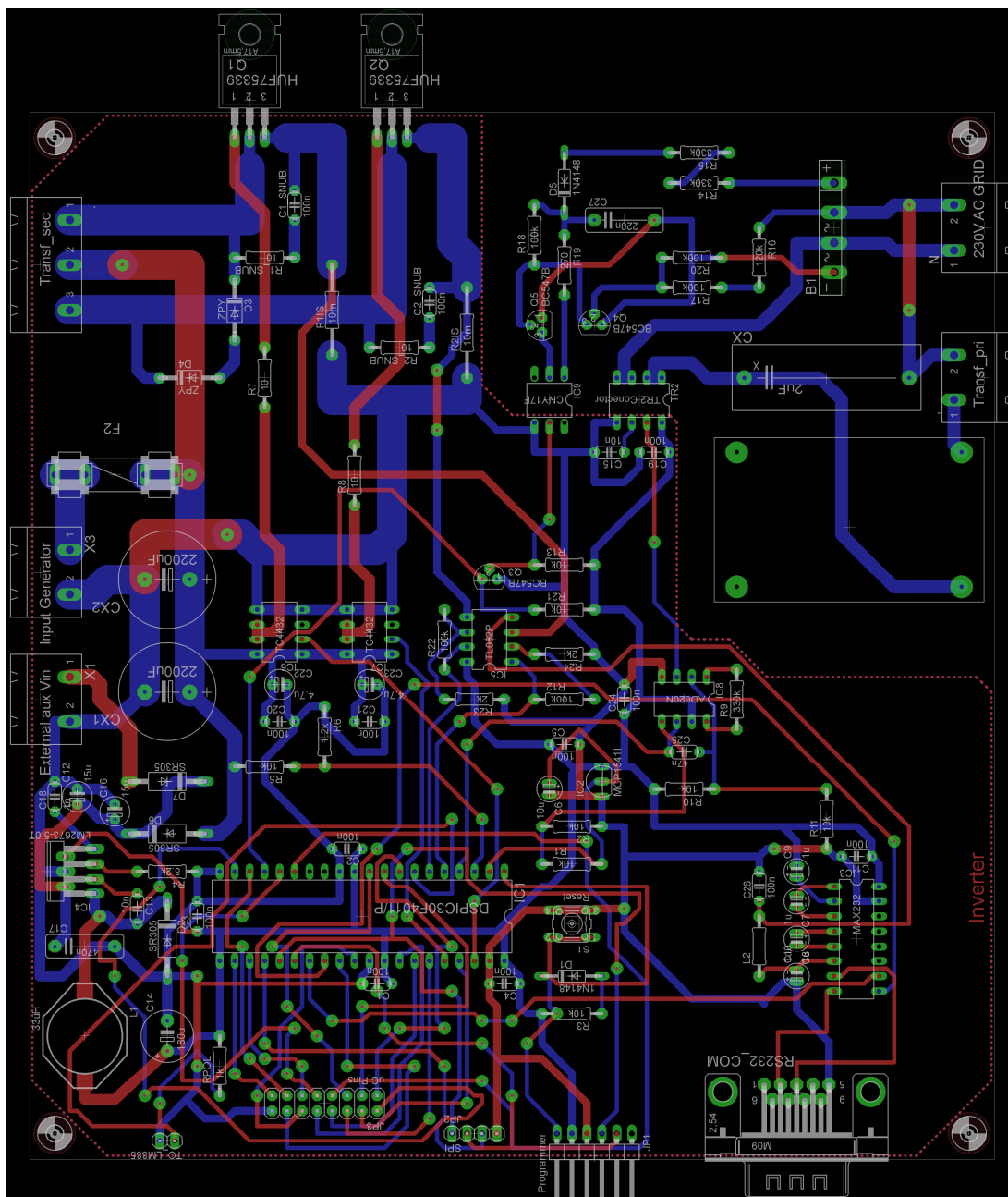


Figura 5. 10 - Placa de circuito impresso

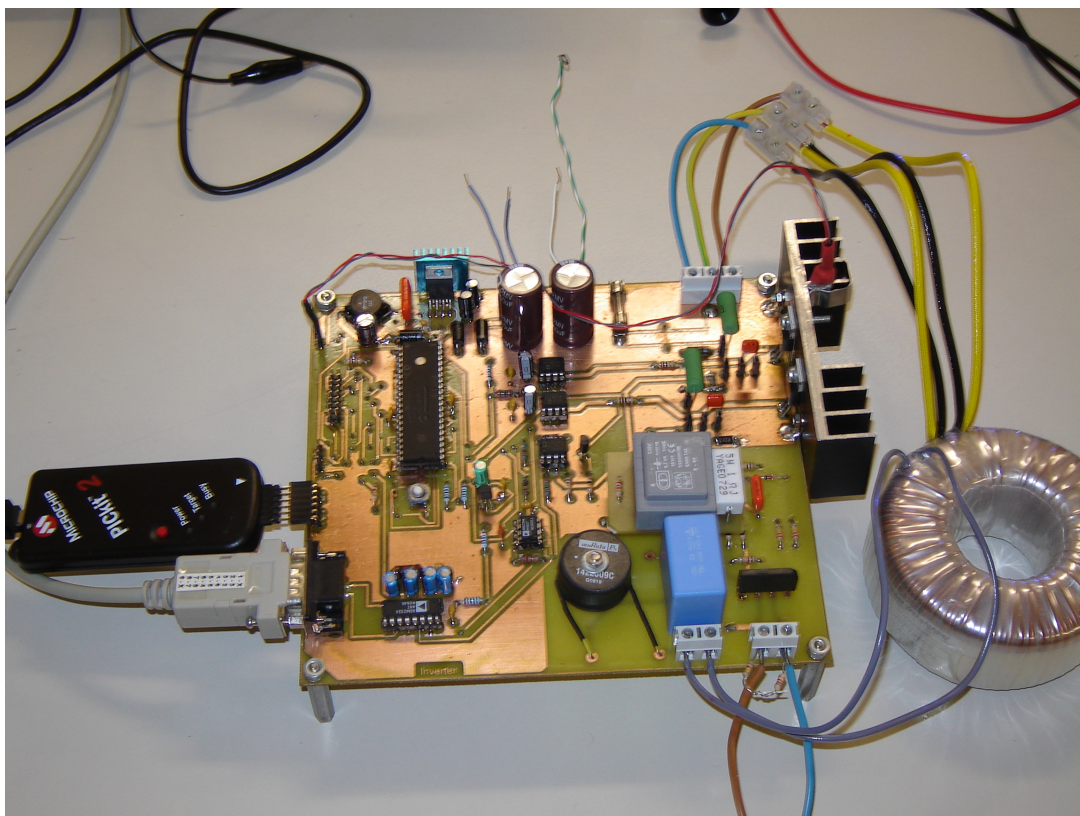


Figura 5. 11 - Aspecto final do circuito inversor

Na Fig. 5.10 encontra-se o desenho da placa de circuito impresso de dupla face com a disposição dos componentes. Como se pode verificar a parte de alta tensão está devidamente isolada do resto do circuito. Os transístores de potência (Q1 e Q2) são montados num dissipador assim como o sensor de temperatura. Foram desenhadas pistas mais largas nas ligações entre o gerador, os transístores Q1 e Q2 e os conectores que ligam os enrolamentos primários de TR1 de modo a poderem comportar correntes mais elevadas.

6. Apresentação de Resultados

6.1. O Software

Uma vez concluída toda a parte de construção e verificação do hardware, procedeu-se a uma série de ensaios tendo em vista avaliar o comportamento do inversor e da máquina de exercício. Todos os testes conduzidos nesta fase basearam-se no funcionamento do inversor em malha aberta e em modo *stand-alone*, onde foram aplicadas cargas com diferentes potências (neste caso várias lâmpadas de 230V com filamento de tungsténio e casquilho E14).

O controlo em malha aberta consiste num sistema que não possui realimentação. O sinal de controlo é aplicado na entrada do sistema esperando-se que à saída a variável de controlo apresente um determinado comportamento desejado. Neste tipo de sistema a informação retirada da evolução do processo não é usada para determinar o sinal de controlo, pelo que a entrada não depende da saída. No caso do inversor deste trabalho, o sinal de controlo é o sinal de SPWM aplicado nas gates de Q1 e Q2 e a variável de saída será a tensão sinusoidal de 50Hz aplicada aos terminais da carga (que possui uma característica puramente resistiva). A amplitude máxima da tensão de saída depende do *duty-cycle* máximo permitido para o sinal de SPWM.

O algoritmo de controlo implementado no dsPIC encarrega-se de comandar o inversor em malha aberta. No fundo o que se faz é percorrer uma tabela de 50 pontos com os valores de seno entre 0° e 180° (meio ciclo). Cada ponto determina o valor do *duty-cycle* a aplicar no PWM em cada instante. Durante os primeiros 10ms a tabela é percorrida com base num *timer* (T1) que gera uma interrupção de alta prioridade a cada 200us. A rotina de atendimento à interrupção actualiza o *duty-cycle* de PWM que é aplicado num dos transístores (por exemplo Q2) de forma a modular a arcada positiva na saída, e incrementa um contador que determina a próxima fase na tabela. No momento em que se chega ao fim da tabela, essa rotina encarrega-se de redireccionar e actualizar o sinal de PWM para o outro transístor (neste caso Q1 e Q2 é desligado) e faz o reset do contador. Nesta fase a arcada negativa é modulada na saída. Ao fim de mais 10ms todo o processo volta a repetir-se.

Foi também implementado um sistema de comunicação série, baseado no protocolo RS-232, que aceita comandos de controlo a partir de um PC. É possível seleccionar o nível de amplitude de saída máxima entre 0% e 100%, ligar ou desligar o inversor e receber periodicamente as leituras obtidas nos canais da ADC. Estes comandos podem ser accionados a partir do HyperTerminal do Windows, ou então a partir de uma pequena aplicação Win32 feita em C# com a ajuda do programa Visual Studio 2008 (ver Fig. 6.1). Os *device drivers* do inversor foram construídos com base em rotinas incluídas nas

livrarias de periféricos para dsPIC's, que o compilador C30 da Microchip disponibiliza e que podem ser consultadas no anexo G (pág. 112). Esta solução reduz consideravelmente o tempo necessário no desenvolvimento de software para este circuito. Toda programação elaborada nesta fase teve como objectivo testar apenas o funcionamento e desempenho dos principais elementos do circuito. Com isto pretendeu-se extrair informações importantes tendo em vista o reajuste e optimização de alguns desses elementos, como se verá mais adiante. No anexo E (pág. 105) pode-se consultar o código fonte da aplicação que corre no dsPIC (inversor_main.c) assim como os device drivers (periféricos.c). No anexo F (pág. 110) está o código fonte da aplicação da Fig. 6.1.



Figura 6. 1 - Detalhe da aplicação de controlo feita em Visual Studio 2008

6.2. Teste de Funcionamento

Como já foi referido anteriormente, os ensaios deste inversor basearam-se no seu funcionamento em malha aberta e em modo *stand-alone*. Usaram-se na sua entrada duas fontes de alimentação DC em paralelo de modo a garantir mais corrente (cada uma fornece até 3A o que dá um total de 6A). A frequência de comutação foi de 20KHz. Segue-se a apresentação das formas de onda dos elementos mais relevantes deste circuito.

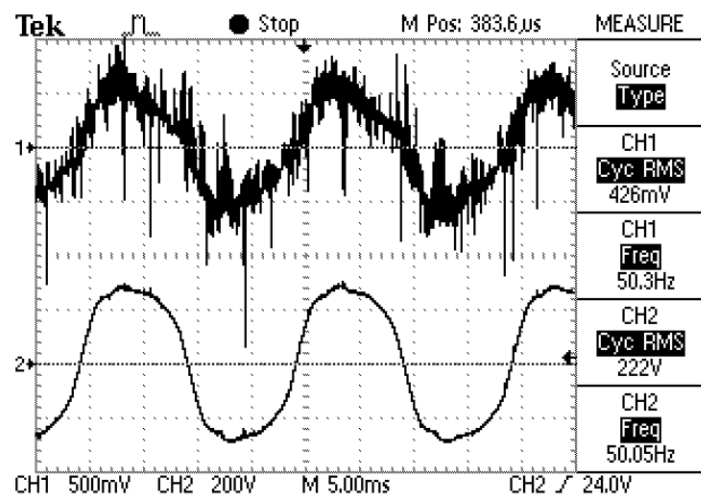


Figura 6. 2 - Onda de corrente (CH1) e tensão (CH2) para $V_{dc}=15V$, $f=20KHz$, carga=25W

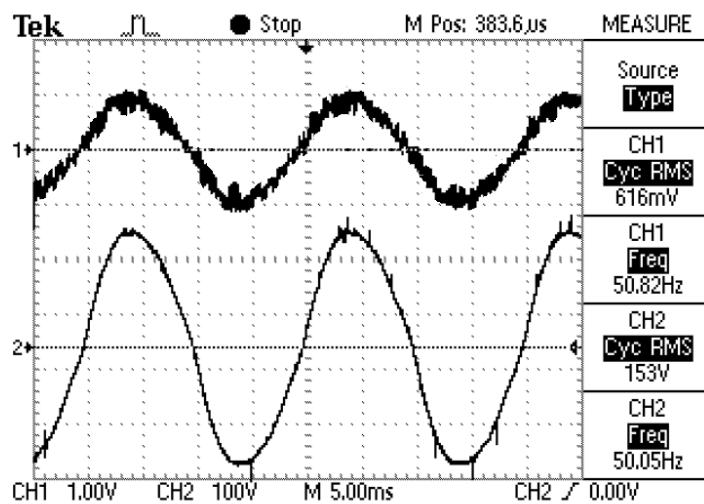


Figura 6. 3 - Onda de corrente (CH1) e tensão (CH2) para $V_{dc}=15V$, $f=20KHz$, carga=60W

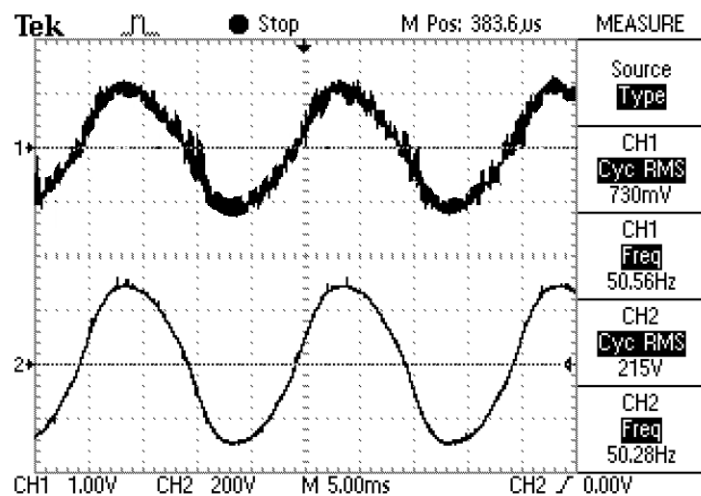


Figura 6. 4 - Onda de corrente (CH1) e tensão (CH2) para $V_{dc}=20V$, $f=20KHz$, carga= 60W

Na Fig.6.2 pode ver-se o resultado da onda de tensão (CH2) e corrente (CH1) de saída do inversor com uma lâmpada de 25W. A tensão de entrada é de 15V e o *duty-cycle* máximo permitido para o SPWM é de 90%. A corrente é medida à saída de IC8 (ver esquema da Fig. 5.8) com o canal 1 em acoplamento AC, pelo que o offset DC de 2.05V é retirado do sinal. O ganho total do andar de acondicionamento do sinal de corrente é de 3.57 (obtido a partir de medições), o que significa que o valor da corrente na realidade é de $426/3.57 = 119\text{mA}_{\text{rms}}$. A tensão é medida aos terminais da carga. Verifica-se que a frequência ronda os 50Hz e o desfasamento entre a corrente e a tensão é praticamente imperceptível, pelo que o $\cos(\varphi) = 1$. De facto estamos perante uma carga que tem um comportamento resistivo. Nota-se uma certa distorção na forma de onda da tensão, isto é, observa-se que não é uma sinusóide totalmente pura (as zonas de amplitude máxima ficam mais arredondadas do que seria suposto). Relativamente à onda de corrente, devido à sua baixa amplitude, a relação sinal/ruído é bastante fraca. No entanto, para amplitudes superiores, esta situação tende a melhorar, conforme se pode observar na Fig. 6.3. Neste teste usou-se uma lâmpada de 60W, para a mesma tensão de entrada. O *duty-cycle* máximo permitido para o SPWM estava em torno dos 75%. A onda de corrente (CH1) e de tensão (CH2) apresentam uma distorção inferior quando comparadas com a imagem anterior. Por outro lado verifica-se uma melhoria significativa na relação sinal/ruído da corrente. Na fig. 6.4 é possível observar a corrente e tensão de saída para uma tensão de entrada de 20V com a carga de 60W para os mesmos 75% de *duty-cycle* máximo. Desta vez a tensão (CH2) chega próxima dos 220Vrms. Para um *duty-cycle* máximo acima de 75% observa-se que a onda de tensão fica ligeiramente distorcida à semelhança do que acontece na Fig. 6.2. Este comportamento é independente da carga ou da tensão de entrada (considerando apenas tensões DC acima de 15V). É preciso não esquecer que estes testes foram feitos em malha aberta e, portanto, partindo destes primeiros resultados satisfatórios, tudo indica que um algoritmo de controlo eficaz poderá minimizar significativamente este problema.

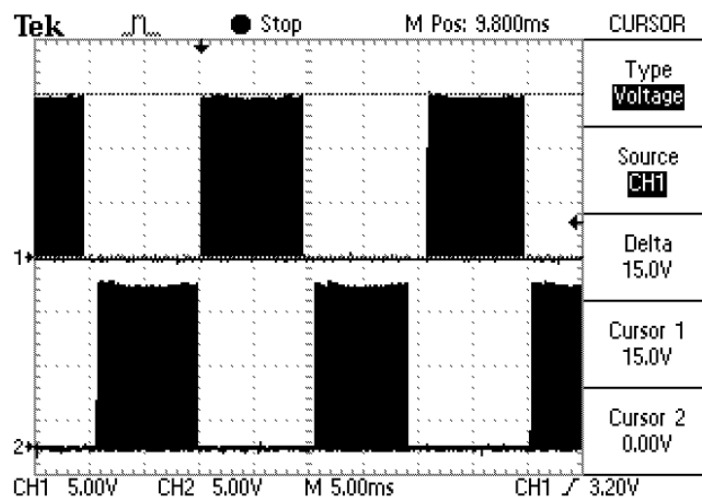


Figura 6. 5 - Sinal de SPWM em Q1 (CH1) e Q2 (CH2) para $V_{dc}=15V$, $f=20KHz$, carga=25W

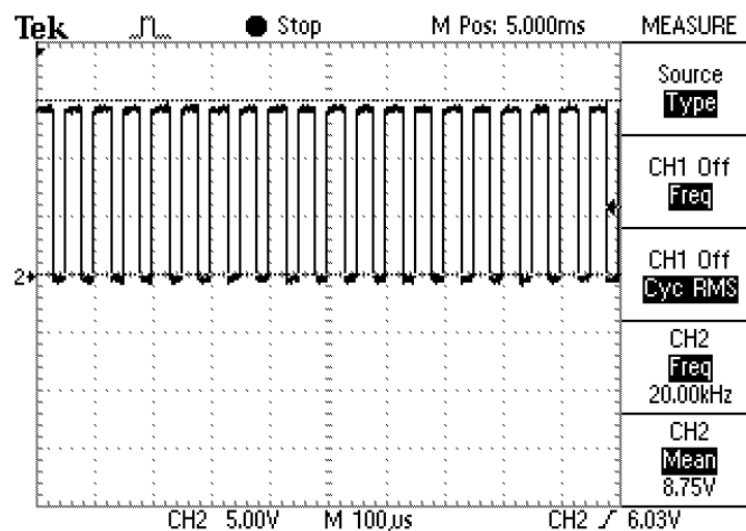


Figura 6. 6 - Detalhe do sinal de SPWM em Q1 (CH2) para $V_{dc}=15V$, $f=20KHz$, carga=25W

Continuando com a análise do funcionamento do inversor, a Fig. 6.5 mostra o sinal de SPWM aplicado em Q1 (CH1) e Q2 (CH2), para uma tensão de entrada de 15V, uma lâmpada de carga de 25W e 75% de *duty-cycle* máximo. É possível observar com um pouco mais de detalhe o sinal de SPWM aplicado a Q1 na Fig. 6.6. A frequência de comutação medida é de 20KHz, como se pode confirmar.

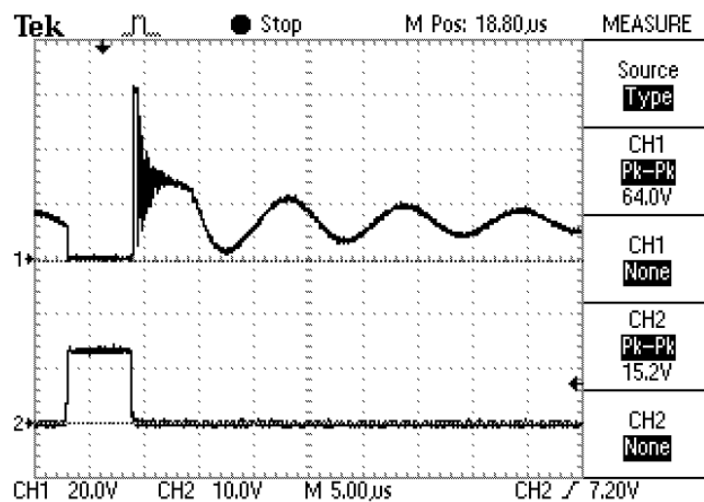


Figura 6. 7 - Tensão de dreno (CH1) e gate (CH2) de Q1 (sem snubber)

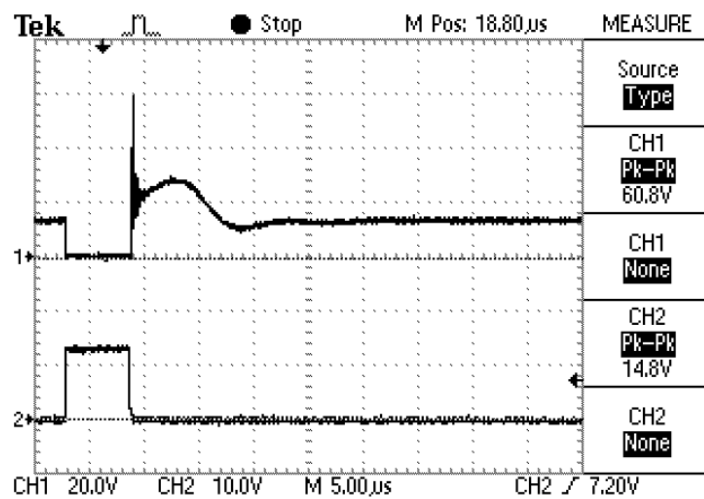


Figura 6. 8- Tensão de dreno (CH1) e gate (CH2) de Q1 (com snubber)

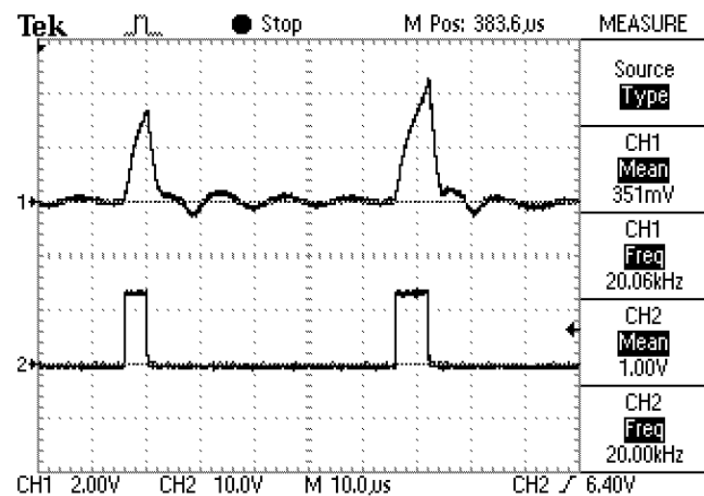


Figura 6. 9 - Corrente de dreno (CH1) e tensão de gate (CH2) em Q1

Na Fig. 6.7 encontra-se o efeito em Q1, descrito no capítulo 3.4 (comparar com a Fig. 3.31). O canal 1 é a tensão de dreno de Q1 e o canal 2 é o respectivo pulso aplicado na gate para $V_{dc} = 15V$, $f = 20KHz$ e carga de 60W. O transistor não tem qualquer tipo de *snubber* aplicado entre a fonte e o dreno. Para desenhar um *snubber* RC que minimize o comportamento da Fig. 6.7, procedeu-se da seguinte forma: começou-se por usar apenas um condensador de 10nF entre a fonte e dreno de Q1. De seguida foi-se aumentando gradualmente o seu valor até que o efeito de *ringing* desaparecesse (e tendo em consideração a Eq. 3.23). Depois acrescentou-se uma resistência (22 ohm) em série com o condensador e foi-se diminuindo gradualmente o seu valor com o objectivo de minimizar o pico de tensão. Obteve-se um valor de 100nF para o condensador e 4.7 ohm para a resistência. O resultado da aplicação deste *snubber* em Q1 encontra-se representado na Fig. 6.8. O canal 1 representa a tensão de dreno de Q1 e o canal 2 o pulso que lhe é aplicado. Verifica-se uma melhoria significativa em termos de *ringing* e pico de tensão.

A Fig. 6.9 mostra o detalhe da corrente que atravessa Q1 (CH1), para dois pulsos de gate (CH2) com larguras diferentes. A corrente foi medida à saída de IC5A (ver esquema da Fig. 5.8) e as condições de teste são iguais às do parágrafo anterior. O ganho total do andar de acondicionamento do sinal de corrente em Q1 é dado por $R1IS \times (1+R12/R24) = 0.51$. No caso do primeiro pulso da Fig. 6.9, o pico de corrente será $3.2V \times 0.51 = 6.3A$. com base na figura, percebe-se que a corrente aumenta à medida que o *duty-cycle* do sinal de gate vai crescendo. Outro detalhe interessante é que a corrente de Q1 não se anula instantaneamente no instante em que o transistor entra ao corte. O que acontece é que a energia armazenada na indutância de fugas de TR1 vista do primário é dissipada através do *snubber* e também através de Q1. Na Fig. 6.8 observa-se que o pico de tensão ultrapassa os 55V (limite máximo para a tensão de disrupção dreno-fonte de Q1) e portanto Q1 conduz ainda alguma corrente logo após a comutação para off.

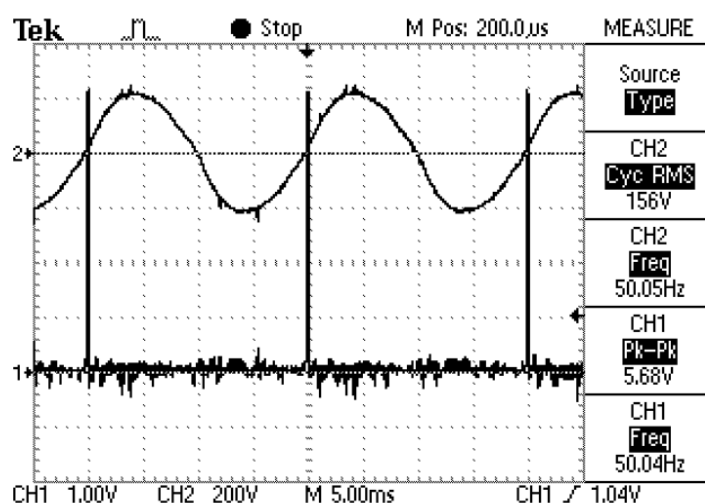


Figura 6. 10 - Tensão de saída (CH2) e detector de passagem por zero (CH1)

Na Fig. 6.10 é possível observar o funcionamento do detector de passagem por zero medido no pino 17 de IC1 (ver Fig. 5.6). Notam-se claramente pulsos de aproximadamente 5V (CH1) apenas numa das passagens por zero da tensão de saída (CH2), conforme o que se tinha idealizado inicialmente. A principal vantagem deste detector é a possibilidade de obter simultaneamente informação sobre o instante em que houve uma passagem por zero e a fase em que ocorreu.

6.3. Desempenho

Neste ponto do trabalho avalia-se o desempenho do inversor em malha aberta ligado ao rolo de treino sob ponto de vista qualitativo. Faz-se também uma avaliação quantitativa ao nível da eficiência do dispositivo para várias condições de funcionamento.

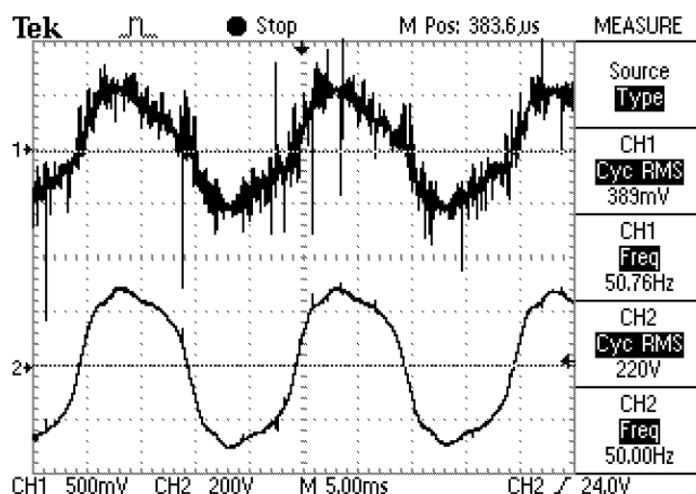


Figura 6. 11 - Onda de corrente (CH1) e tensão (CH2) para V_{dc} =gerador, f =20Khz, carga=25W

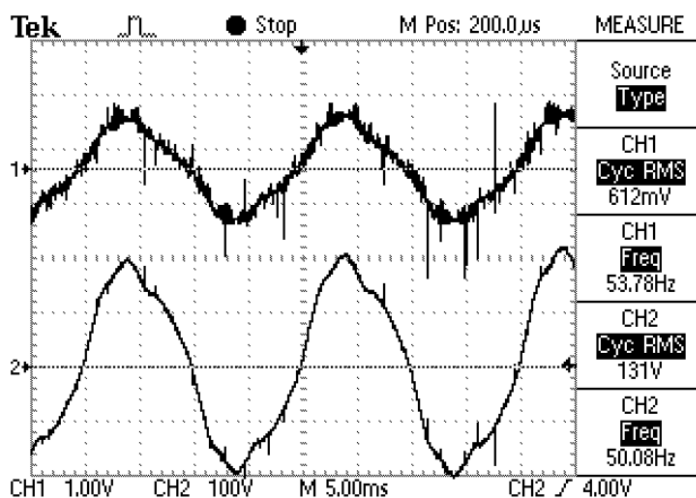


Figura 6. 12 - Onda de corrente (CH1) e tensão (CH2) para V_{dc} =gerador, f =20Khz, carga=60W

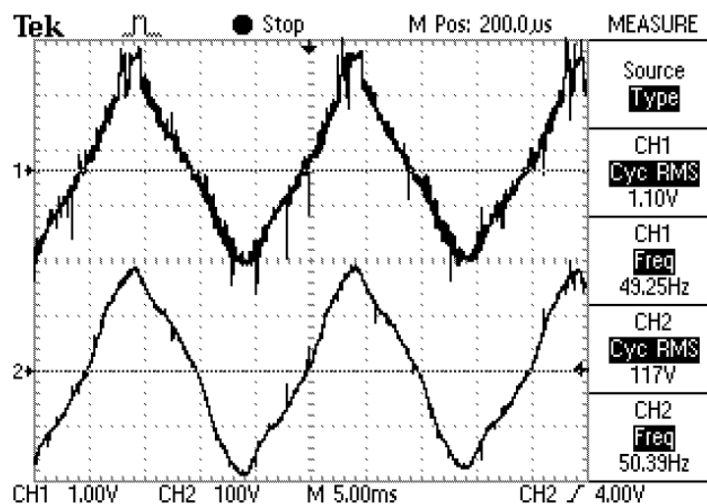


Figura 6. 13 - Onda de corrente (CH1) e tensão (CH2) para V_{dc} =gerador, f =20Khz, carga=100W

As figuras 6.11, 6.12 e 6.13 mostram as formas de onda da corrente (CH1) e da tensão (CH2) de saída para cargas diferentes. O *duty-cycle* máximo permitido para o SPWM era de 100%. Quando o gerador está parado, a alimentação auxiliar do circuito é assegurada através de um pequeno adaptador DC de 12V. O comportamento do inversor revela-se muito semelhante ao que acontecia quando estava ligado às fontes de alimentação DC. A principal diferença reside no facto da tensão de entrada não ser constante. É muito difícil para uma pessoa manter um ritmo de movimentos estável no rolo de treino e essa situação reflecte-se no inversor. A lâmpada de carga cintilava durante o exercício e isso notava-se na amplitude irregular da tensão de saída. No entanto será possível minimizar este efeito com um algoritmo de controlo em malha fechada que regule eficazmente a potência de saída (ao mesmo tempo que melhora a distorção visível, por exemplo, na Fig. 6.13). Outra característica interessante observada quando se aumenta gradualmente a potência de saída (variando o *duty-cycle* máximo) é que o nível de esforço físico sobe consideravelmente. Sente-se nos pedais uma força que contraria o acto de pedalar. Isto demonstra que é possível regular o nível de esforço do aparelho de exercício sem a necessidade dos dispositivos de atrito referidos no capítulo 2.1, usando apenas a energia gerada (seleccionando uma determinada potência de saída).

Para medir a eficiência do inversor foram novamente ligadas as fontes DC na sua entrada juntamente com um amperímetro digital. Foi ligado também um amperímetro digital na saída em série com a carga. O *duty-cycle* máximo programado para o SPWM não ia além dos 75%. As medições obtidas para diferentes condições de funcionamento encontram-se nas seguintes tabelas:

Lâmpada	15W	25W	60W	100W	125W
Vdc (V)	15	15	15	15	15
Idc (A)	2.2	2.85	3.79	4.53	4.78
Vac (V_{rms})	220	200	155	103	80.5
Iac (mA_{rms})	70	120	220	300	340
η (%)	46.7	56.1	59.9	45.5	38.2

Tabela 6. 1 - Eficiência em função da carga de saída

Vdc (V)	20	17	15	13	10
Idc (A)	2.75	2.92	3.13	3.29	3.72
Vac (V_{rms})	124	124	127	125	126
Iac (mA_{rms})	180	180	180	180	180
η (%)	40.6	44.9	48.7	52.6	60.9

Tabela 6. 2 - Eficiência em função da tensão de entrada com lâmpada de 60W

f (KHz)	Onda quadrada a 50Hz	5	10	20	58	80
Vdc (V)	15	15	15	15	15	15
Idc (A)	3.5	3.26	3.14	3.05	3.30	3.53
Vac (V_{rms})	173	125	124	124	124	124
Iac (mA_{rms})	240	180	180	180	180	180
η (%)	79.1	46.0	47.4	48.8	45.1	42.2

Tabela 6. 3 - Eficiência em função da frequência de comutação com lâmpada de 60W

Onde:

Vdc - Tensão de entrada no inversor

Idc - Corrente de entrada no inversor

Vac - Tensão de saída no inversor

Iac - Corrente de saída no inversor

η - Eficiência calculada em que $\eta = (Vac \times Iac \times 100) / (Vdc \times Idc)$

f - Frequência de comutação

A tabela 6.1 mostra a eficiência do inversor para diferentes cargas. A frequência de comutação neste caso é de 20KHz. O melhor resultado é obtido para uma carga equivalente a uma lâmpada de 60W.

A tabela 6.2 reflecte o resultado da eficiência quando se varia a tensão de entrada. Verifica-se que quanto menor for a tensão de entrada, melhor será o rendimento do inversor.

A tabela 6.3 revela a eficiência do dispositivo em função da frequência de comutação. Foi também criada uma pequena rotina para se poder modular uma onda quadrada a

50Hz com um *duty-cycle* de 40% (apenas para comparação de resultados). Neste modo de funcionamento a eficiência atinge quase os 80%. Os inversores *stand-alone* de baixo custo, abordados no capítulo 2.2, trabalham precisamente desta forma. Para o modo SPWM, o melhor resultado obtém-se para frequências mais baixas (entre os 5KHz e os 20KHz). Acima deste intervalo a eficiência diminui bastante com o aumento da frequência de comutação. Este resultado está de acordo com o que foi referido no ponto 3.2.4 (recordar as equações 3.7 e 3.8), uma vez que se está a utilizar um transformador toroidal com núcleo laminado de ferro. As perdas por histerese e por correntes de Foucault são directamente proporcionais à frequência e isto traduz-se num aquecimento considerável no transformador. Apesar de se estar a modular uma sinusóide a 50Hz, verifica-se que a componente de alta-frequência tem uma influência significativa nas perdas observadas. Abaixo dos 5KHz o filtro de saída LC deixa de ser eficaz, obrigando ao seu redimensionamento a nível da frequência de corte (terá de se baixar). Por outro lado a componente de 5KHz encontra-se já na banda audível, e de facto ouve-se um zumbido desagradável vindo do transformador. Isto não é mais do que vibrações mecânicas (entram em ressonância a 5KHz) que acontecem no laminado que compõe o núcleo de TR1.

7. Conclusões e Trabalhos Futuros

Este trabalho focou-se no estudo, desenvolvimento e construção de um rolo de treino com gerador DC e um inversor baseado na topologia push-pull.

Com o auxílio de um simulador, comprovaram-se os resultados do inversor nessa topologia. Partindo da modulação em SPWM, conseguiu-se obter uma onda de corrente sinusoidal programável e em fase com a rede, usando como entrada um gerador DC de baixa tensão. Demonstrou-se também ser possível debitar potência directamente na rede ao mesmo tempo que se garantiu estabilidade no processo de controlo em malha fechada com base num compensador PID. Constatou-se que uma parte significativa da energia era dissipada nos transístores e restantes componentes o que resultou numa eficiência em torno dos 45.7%, no melhor dos casos. A simulação permitiu também fazer uma análise detalhada do funcionamento geral do inversor o que possibilitou a rápida correcção de pequenos problemas entretanto detectados no circuito. É por isso uma ferramenta flexível que reduz consideravelmente o tempo gasto no desenvolvimento de circuitos electrónicos.

O rolo de treino com gerador DC construído para este trabalho reforça a ideia de que é possível substituir os mecanismos de atrito dos aparelhos cardiovasculares por outros que aproveitam de facto parte da energia gerada. Os ajustes desenvolvidos neste protótipo permitiram que se possa aplicar qualquer tipo de bicicleta com um baixo custo. Algumas alterações poderão ser feitas com o objectivo de minimizar as perdas mecânicas. Por exemplo, se o eixo do motor for ligado directamente ao eixo do rolo, evitam-se os atritos provocados pelo uso de uma correia dentada. Os testes com carga revelaram que é difícil manter uma rotação estável no rolo, principalmente nos instantes em que a força exercida muda de um pedal para o outro. A aplicação de uma roda de inércia com uma massa superior àquela que foi implementada, poderá permitir uma melhor estabilidade ao nível da rotação do rolo. Para um nível de esforço físico elevado, gerou-se com este aparelho 48W de potência.

Os testes em malha aberta efectuados no circuito inversor produziram resultados satisfatórios. Nesta configuração foi possível gerar uma tensão sinusoidal a 50Hz e alimentar várias cargas baseadas em lâmpadas de diferentes potências. A modulação por SPWM demonstrou ser uma boa solução na base de um inversor true-sinewave. O bom funcionamento do detector de passagem por zero aliado a um algoritmo de controlo em malha fechada permitirá obter ondas de saída (corrente e tensão) síncronas com a rede, com menores distorções. A capacidade de regulação na saída também sairá beneficiada. Os dados obtidos para a eficiência do inversor mostram que o transformador TR1 é o principal responsável pelas perdas verificadas. Constatou-se que a eficiência depende da tensão de entrada, da potência de saída e, fundamentalmente, da frequência

de comutação. Quanto maior for a frequência de switching, pior será o desempenho do inversor a nível de eficiência. As perdas de TR1 resultam no seu aquecimento. O modelo de simulação usado para o bloco do transformador não teve em linha de conta as perdas no núcleo. Ao contrário do que foi observado na simulação, os transístores de potência acabaram por dissipar menos energia do que seria de esperar. Uma solução para se poder melhorar a eficiência seria usar um transformador em que o núcleo fosse constituído por lâminas de ferro mais finas (por exemplo 0.3 mm em vez de 0.5 mm de espessura) reduzindo as perdas por correntes de Foucault.

A utilização do aparelho de exercício ligado ao inversor revelou a capacidade de gerar energia e alimentar uma carga independente. De um modo geral, pode dizer-se que neste trabalho foram atingidos objectivos importantes que confirmaram a viabilidade de incluir nos aparelhos de exercício físico comerciais um sistema de aproveitamento energético de baixo custo. Nesta perspectiva, o exercício físico é também uma potencial fonte de energia renovável.

A etapa seguinte no desenvolvimento deste projecto será a implementação de software de controlo e monitorização em modo grid-connected do inversor. Também será necessário incluir um display LCD e botões de comando que servirão de interface com o utilizador. O nível de esforço físico no aparelho de exercício poderá ser programado em função da quantidade de energia que é inserida na rede, entre outras funcionalidades. Os resultados obtidos assim como o software desenvolvido para este trabalho funcionam como uma plataforma que permitirá dar continuidade a este projecto.

Referências

- [1] William M. Flanagan, “Handbook of Transformer Design & Applications - 2nd Ed.”, McGraw-Hill, 1993
- [2] Alexandre Manuel Mota, “Electrónica de Potência”, DETI, Universidade de Aveiro, 2006
- [3] Apontamentos Teóricos da Disciplina de Electrotecnia Teórica, DETI, Universidade de Aveiro, 2005
- [4] U.S. Department of Energy, “DOE Fundamentals Handbook Vol. 2”, Washington D.C., 1992
- [5] O. A. Mohamed, “Standard Handbook of Electrical Engineers”, Section 8, McGraw-Hill, 2006
- [6] Bruno P. N. Pincante, “Microgeração”, Departamento de Engenharia Electrotécnica e de Computadores, Universidade de Coimbra, 2007
- [7] Timothy L. Skvarenina, “The Power Electronics Handbook”, CRC Press, 2002
- [8] Alberto C. A. Santander, “Inversor Push-Pull a Três Níveis”, Universidade Federal de Santa Catarina, 1993
- [9] Ilias Tsagas, “Laboratory Evaluation of DC/AC Inverters for Stand-alone & Grid-connected Photovoltaic Systems”, University of Strathclyde, 2002
- [10] Ned Mohan, “First Course on Power Electronics and Drives”, MNPERE, 2003
- [11] Muhammad H. Rashid, “Power Electronics Handbook”, Academic Press, 2001
- [12] Ricardo J. S. Matias, “Inversor 24V/230V para Sistemas Domésticos de Produção de Energia Eólica”, DETI, Universidade de Aveiro, 2007
- [13] Rudy Severns, “Design of Snubbers for Power Circuits”
<http://www.cde.com/tech/design.pdf>

[14] Site de equipamentos de exercício da SportsArt,
<http://www.sportslinq.com/cycles/index.html>

[15] Site de equipamentos de exercício da Stex
<http://www.stexfitness.com/com-1.html>

[16] Site de sistemas de energia da SMA Solar Technology
<http://www.sma.de/>

[17] Site sobre energias renováveis
<http://www.renovaveisnagora.pt/entrada>

[18] Site da OrCAD Community
<http://www.orcad.com/community.pspice.faq.aspx#modeling>

[19] Site da Microchip
<http://www.microchip.com/>

Anexo A – Bicicleta cardiovascular da Stex



STEX 8000 Recumbent Bike

Our Recumbent Bikes come with an ergonomically designed backrest that keeps your body in an ideal and comfortable position, all through the workout. Besides this, they also feature a mobile phone holder and a magazine rack. So you make the most of your exercise routine. Each and every time.

STEX 8020R

STEX 8020RL



▲ Multilingual LCD Window
Firmware Upgrade RU-46 port

▲ 17" Attachable TFT LCD
Multilingual LCD Window
Display Tilt available
Integrated Volume / Channel Buttons
Watch TV, VTR, VCD, DVD, PMP
Headphones, Headset through Audio-Out

▲ Magazine rack, Cup holder and
Mobile phone holder

▼ Intelligent Polar Telemetry / Built-in handgrip sensor that enables interactive heart rate programs.
Seat, handlebars aligned so as to provide maximum ergonomic support and mobility.



1. Self-powered Alternator System

Self-powered, so you needn't worry about cords and outlets. Self-contained alternator brake system, to control the pace at which you exercise.

2. Adjustment for the seat

Sliding, locking seat adjustment with 12 different positions.

4. Ergonomic drive system

Belt drive system for a smooth and effortless fitness session. Heavy duty frame with a double triangular structure, for reinforced strength and durability.



Attachable LCD model

1. Integrated Volume / Channel Buttons
Headphones, Headset through Audio-out
2. Display Tilt available



SPECIFICATIONS

8020R

Driving System

- Electrical resistance control
- Self-powered supply by alternator

Display

- Multilingual LCD window
- Firmware upgrade RJ-45 port
- Smart card access device (optional)

Workout Program

- 6 Preset programs
- 4 Interactive HRC programs
- 3 Manual programs
- User defined programs
- Fitness Test
- Military Fitness Test
- Shortcut buttons for HRC FAT BURN & HRC CARDIO
- Resistance levels : 25

Heart Rate

- Polar® telemetry (chest strap required-optional)
- Intelligent heart rate system (HRS)® touch H/R plus

Dimension

- 68(W), 123(L), 135(H) cm
- 60kg
- Resistance 20-500 watt

Warranty

- 3 Years for all components
- Lifetime for steel frame

8020RX, RL

Driving System

- Electrical resistance control
- Extra power supply with AC adaptor (for TFT-LCD Screen)

Display

- Multilingual LCD window
- Firmware upgrade RJ-45 port
- 8020RX : 12" Built-in TFT-LCD
- 8020RL : 17" Attached TFT-LCD

Workout Program

- 6 Preset programs
- 4 Interactive HRC programs
- 3 Manual programs (except 8020RX)
- User defined programs (except 8020RX)
- Fitness Test (except 8020RX)
- Military Fitness Test (except 8020RX)
- Shortcut buttons for HRC FAT BURN & HRC CARDIO (except 8020RX)
- Resistance levels : 25

Heart Rate

- Polar® telemetry (chest strap required-optional)
- Intelligent heart rate system (HRS)® touch H/R plus

Dimension

- 8020RX : 68(W), 123(L), 135(H) cm
- 8020RL : 68(W), 123(L), 169(H) cm
- 8020RX : 60kg / 8020RL : 70kg
- Resistance 20-500 watt

Warranty

- 3 Years for all components
- Lifetime for steel frame
- LCD Monitor : 2 Year

Anexo B – Inversor *stand-alone* da SMA Technology



SI 5048 / SI 5048U

Flexible

- > For systems from 3 to 100 kW
- > Single-, split- and three-phase operation, connectable in parallel and modularly extendable
- > AC and DC coupling

Simple

- > Comfortable commissioning due to "Quick Configuration Guide"
- > Complete off-grid management

Efficient

- > High efficiency
- > Intelligent battery management prolongs battery service life
- > Reliable state-of-charge evaluation

Safe

- > Extreme overload capability
- > OptiCool
- > 5-year SMA warranty



SUNNY ISLAND 5048

Powerful Off-Grid Manager with Innovative Technology

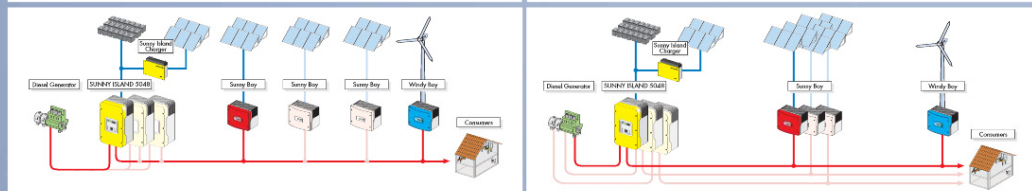
As the central unit of a stand-alone power supply system, battery inverters have many different functions and are often difficult to configure. Yet the Sunny Island 5048 from SMA shows that simplicity is also possible. Fit for systems of 3 to 100 kW, it can be commissioned in a matter of a few minutes. All settings required for operation can be made quickly and simply. The Sunny Island 5048 is flexible in its application, extendable and takes on all control processes. Its first-class battery management ensures the maximum service life of the batteries protecting your investment. In addition, the Sunny Island 5048 features many of the innovative product characteristics which made SMA a winner in a test carried out recently by the German product standards institute "Stiftung Warentest": high efficiency, an ergonomic aluminum housing and its OptiCool ventilation system. The Sunny Island is also available in an UL-compliant 5048U version with an output rating of 120 V and 60 Hz.

Technical Data

SUNNY ISLAND 5048 / 5048U

	SI 5048	SI 5048U
Output data		
Nominal AC voltage (adjustable)	230 V (202 - 253 V)	120 V (105 - 132 V)
Nominal grid frequency (adjustable)	50 / 60 Hz (45 - 65 Hz)	60 Hz (55 - 65 Hz)
Continuous AC power at 25 °C / 45 °C	5000 / 4000 W	5000 / 4000 W
Continuous AC power at 25 °C for 30 / 5 / 1 min	6500 / 7200 / 8400 W	6500 / 7200 / 8400 W
Max. AC power for 3 s	12000 W	11000 W
Nominal AC current	21.7 A	41.7 A
Max. AC current	120 A (60 ms)	180 A (60 ms)
Output voltage harmonic distortion factor	< 3 %	< 3 %
Power factor	-1 to +1	-1 to +1
Input data		
Input voltage (range)	230 V (172.5 - 250 V)	120 V (80 - 150 V)
Input frequency	50 / 60 Hz (40 - 70 Hz)	60 Hz (54 - 66 Hz)
Max. AC input current (adjustable)	56 A (2 - 56 A)	56 A (2 - 56 A)
Max. input power	12.8 kW	6.7 kW
Battery data		
Nominal battery Voltage (range)	48 V (41 - 63 V)	48 V (41 - 63 V)
Max. battery charging current	120 A	120 A
Continuous charging current	100 A	100 A
Battery capacity	100 - 10 000 Ah	100 - 10 000 Ah
Charge control	IU ₀ U with automatic full and equalization charge	IU ₀ U with automatic full and equalization charge
Efficiency/power consumption		
Max. efficiency (typical)	95 %	95 %
Own consumption with no load (standby)	25 W (< 4 W)	25 W (< 4 W)
Protection type (DIN EN 60529)	IP30	NEMA 1
Certification	CE	UL
Device protection	short-circuit, overload, overtemperature	short-circuit, overload, overtemperature
Interfaces	2 LEDs, 4 buttons, 2-line display, 2 multifunction relays, RS485, SD card	2 LEDs, 4 buttons, 2-line display, 2 multifunction relays, RS485, SD card
Mechanical data		
Width / height / depth	467 / 612 / 235 mm	467 / 612 / 235 mm
Weight	63 kg	63 kg
Ambient conditions		
Ambient temperature	-25 °C ... +50 °C	-25 °C ... +50 °C
Warranty (EU)	5 years	5 years
Accessories		
Ext. battery temperature sensor	included	included
"GenMan" generator start manager	optional	optional
Multiclusster-Box	optional	-
"BatFuse" battery connection box	optional	optional

Version: December 2008



www.SMA.de
 Freecall +800 SUNNYBOY
 Freecall +800 78669269

SMA Solar Technology AG

5048EN084912 SMA and Sunny Island are registered trademarks of SMA Solar Technology AG. Text and figures comply with the state of this art applicable when printing. Subject to technical changes. Printed on chlorine-free paper.

Anexo C – Inversor *grid-connected* da SMA Technology



SB 2000HF-30 / SB 2500HF-30 / SB 3000HF-30

High Yields

- > Maximum efficiency over 96 %
- > Best tracking efficiency with OptiTrac MPP control
- > OptiCool active temperature management

Safe

- > Galvanic Isolation
- > Integrated DC load disconnecting unit ESS
- > Anti-theft protection

Easy to use

- > Fast and easy installation due to plug-in communication unit
- > Innovative DC plug system SUNCLIX
- > Plug-in grounding (optional)

Communicative

- > Simple country configuration
- > Multi-language graphic display
- > Bluetooth® technology



SUNNY BOY 3000HF

Simply high yields

This is the new generation of galvanically isolated inverters – packed full of the latest SMA technology, the Sunny Boy HF series provides the highest yields for transformer inverters of this power class. Installation is now easier than ever thanks to the new DC connector system SUNCLIX, the plug-in generator grounding (optional), and an easily accessible configuration area – including easier mounting due to the reduced weight. The wide input voltage range from 175 to 700 Volt gives you extraordinary flexibility for your system design. And the modern graphic display and wireless Bluetooth® communication system make the devices highly user-friendly.

Technical Data

SUNNY BOY 2000HF / 2500HF / 3000HF

	SB 2000HF-30	SB 2500HF-30	SB 3000HF-30
Input (DC)			
Max. DC power	2100 W	2650 W	3150 W
Max. DC voltage	700 V	700 V	700 V
PV voltage range, MPPT	175 V - 560 V	175 V - 560 V	210 V - 560 V
Max. input current	12.0 A	15.0 A	15.0 A
Number of MPP trackers	1	1	1
Max. number of strings (parallel)	2	2	2
Output (AC)			
Nominal AC power / max. AC power	2000 W / 2000 W	2500 W / 2500 W	3000 W / 3000 W
Max. output current	11.4 A	14.2 A	16.0 A
Nominal AC voltage / range	220 V - 240 V / 180 V - 260 V	220 V - 240 V / 180 V - 260 V	220 V - 240 V / 180 V - 260 V
AC grid frequency / range	50 Hz / 60 Hz / ± 4.5 Hz	50 Hz / 60 Hz / ± 4.5 Hz	50 Hz / 60 Hz / ± 4.5 Hz
Power factor (cos φ)	1	1	1
THD	< 4 %	< 4 %	< 4 %
AC connection	single-phase	single-phase	single-phase
Efficiency			
Max. efficiency / Euro-Eta	> 96 % / > 95 %	> 96 % / > 95 %	> 96 % / > 95 %
Protection devices			
DC reverse polarity protection	●	●	●
DC load-disconnecting unit ESS	●	●	●
AC short-circuit protection	●	●	●
Ground fault monitoring	●	●	●
Grid monitoring (SMA Grid Guard)	●	●	●
All-pole-sensitive residual current monitoring unit	●	●	●
General data			
Dimensions (W / H / D) in mm	348 / 580 / 145	348 / 580 / 145	348 / 580 / 145
Weight	< 18 kg	< 18 kg	< 18 kg
Operating temperature range	-25 °C ... +60 °C	-25 °C ... +60 °C	-25 °C ... +60 °C
Consumption: (standby) operation / night	< 5 W / 0.25 W	< 5 W / 0.25 W	< 5 W / 0.25 W
Topology	HF-transformer	HF-transformer	HF-transformer
Cooling concept	n. s.	OptiCool	OptiCool
Installation: indoors / outdoors (IP65 electronics)	●/●	●/●	●/●
Features			
DC connection: SUNCLIX	●	●	●
AC connection: connector	●	●	●
Graphic display	●	●	●
Interfaces: Bluetooth / RS485	●/○	●/○	●/○
Warranty: 5 years / 10 years / 15 years / 20 years / 25 years	●/○/○/○/○	●/○/○/○/○	●/○/○/○/○
Certificates and approvals	www.SMA.de	www.SMA.de	www.SMA.de
<input checked="" type="checkbox"/> Standard equipment <input type="checkbox"/> Optional	Data at nominal conditions – provisional data, as of September, 2009		

Accessories



Plug-in generator grounding connection



Communication unit RS485
+ multi-function relay

Anexo D – Propriedades típicas de alguns materiais magnéticos

TABLE 6.1 Typical Properties of Important Soft Magnetic Materials

Material	Properties							Application Notes	
	Approximate Composition, %	B _s , kG	B _R , kG	H _c , Oe	Core Loss, W/lb	Permeability			
						Initial	Maximum		
Silicon Steels									
0.014 in thick AISI M-6	Fe	97	19	14	0.1	0.66 at 15 kG 60 Hz	350	50,000	Widely used for laminations and I bars at power and audio frequencies. Most effective when flux path is in direction of rolling.
	Si	3							
0.012 in thick AISI M-5	Fe	97	19	14	0.1	0.58 at 15 kG 60 Hz	350	50,000	Used in wound C cores mostly for operation at 50/60 Hz. Has low core losses at high flux densities.
	Si	3							
0.0185 in thick AISI M-19	Fe	97	19	—	0.5	0.80 at 10 kG 60 Hz	300	10,000	Used for laminations when higher core loss is acceptable. Less costly.
	Si	3							
0.025 in thick AISI M-22	Fe	97	19	—	0.6	0.9 at 10 kG 60 Hz	300	10,000	Higher losses and less costly than M-19.
	Si	3							
0.004 in thick Grain-oriented	Fe	97	19	14	0.4	10.0 at 15 kG 400 Hz	350	50,000	Widely used in wound cores cut and uncut at 400 Hz and higher frequencies.
	Si	3							
0.002 in thick Grain-oriented	Fe	97	19	14	0.5	14.0 at 10 kG 1.0 kHz	350	50,000	Used in wound cores cut and uncut for high-frequency and pulse use.
	Si	3							

TABLE 6.1 (continued)

Material	Properties							Application Notes
	Approximate Composition, %	B_s , kG	B_R , kG	H_c , Oe	Core Loss, W/lb	Permeability		
						Initial	Maximum	
Nickel-Iron Alloys								
0.014 in thick 50% Ni	Fe 50	13	11	0.15	3.0 at 10 kG	5,000	40,000	Used in laminations to provide high permeability at high flux densities at audio frequencies.
	Ni 50				400 Hz			
0.006 in thick 50% Ni	Fe 50	13	11	0.15	2.0 at 10 kG	5,000	100,000	Applications similar to 0.014-in-thick material but lower losses at high frequencies. Labor cost of stacking laminations very high.
	Ni 50				400 Hz			
0.014 in thick 80% Ni	Fe 20	7.5	6	0.05	0.8 at 6 kG	30,000	100,000	Used in laminations to provide very high initial permeability. Losses high at high frequencies. Saturates at low flux density.
	Ni 80				400 Hz			
0.006 in thick 80% Ni	Fe 20	7.5	6	0.05	0.4 at 6 kG	30,000	60,000	Applications similar to 0.014-in-thick material with lower losses at high frequencies. Labor cost of stacking laminations very high.
	Ni 80				400 Hz			
0.004 in thick 80% Ni	Fe 20	7.5	6	0.05	8.0 at 6 kG	30,000	60,000	Used in wound cores cut and uncut to provide high permeability and low losses at high frequencies.
	Ni 80				5 kHz			
0.004 in thick Square loop Ni-Fe	Fe 50	15	14.5	0.11	1.3 at 10 kG	—	—	Used in saturating and switching devices. Square hysteresis loop realizable only in toroidal cores.
	Ni 50				400 Hz			

<i>Supermendur*</i>						
0.004 in thick	Fe 51 Co 49	22	21	0.2	14.0 at 20 kG 400 Hz	800 27,000 Used in wound cores providing highest saturation flux density. Very costly.
<i>Metallic Glass</i>						
0.001 in thick	Fe 81 B 13 Si 3.5	16	11	0.06	10.0 at 6 kG 10 kHz	2,500 100,000 Has low losses at high frequencies and high flux densities. Available in cut cores and toroids. Very costly. Under development.
<i>Powdered Alloys</i>						
	Variable Fe Ni Mo	8	—	—	9.0 at 1 kG 20 kHz	200 210 Used in high-frequency high- <i>Q</i> applications. Available in toroids and slugs.
<i>Ferrites</i>						
Mn-Zn	Fe ₂ O ₃ ·MnO Fe ₂ O ₃ ·ZnO	4.5	1.0	0.2	0.05 W/cm ³	2,700 4,800 Used in high-frequency and relatively high flux density applications. Available in pot cores and proprietary shapes.
Ni	Fe ₂ O ₃ ·NiO	3.2	2.6	4.0	—	120 150 Available in pot cores for use at frequencies into the megahertz range.

*Registered trade name of Arnold Engineering Co.

Anexo E – Código fonte do microcontrolador

```
inversor_main.c

#include <p30fxxxx.h>
#include "perifericos.h"
#include <pwm.h>
#include <uart.h>
#include <timer.h>
#include <adc10.h>

#define TRUE 1
#define FALSE 0

_FOSC(CSW_FSCM_OFF & XT_PLL16); //30F4011 Configuration Bits
_FWDT(WDT_OFF);
_FBORPOR(MCLR_EN & PWRT_OFF);
_FGS(CODE_PROT_OFF);

//-----
//Declaração das variáveis globais
static unsigned char arcada_positiva = TRUE, mosfet = 1, phase = 0, amplitude = 0, pwm_constante
= FALSE;

//Os valores para a tabela de seno podem ser programados usando/tirando os
//caracteres de comentários dos respectivos blocos
static unsigned int sin_table[] =

/*{0,231,462,691,917,1139,1357,1569,1776,1975,2167,2350,2523,2687,2840,2982,3112,3230,3335,3427,35
06,3570,3621,3657,3679,3686,3679,3657,3621,3570,3506,3427,3335,3230,3112,2982,2840,2687,2523,2350,
2167,1975,1776,1569,1357,1139,917,691,462,231}; // -> 1KHz pre de 16*/

/*{0,741,1478,2210,2934,3645,4342,5022,5683,6321,6934,7519,8075,8599,9089,9543,9960,10337,10673,10
968,11219,11425,11587,11703,11773,11796,11773,11703,11587,11425,11219,10968,10673,10337,9960,9543,
9089,8599,8075,7519,6934,6321,5683,5022,4342,3645,2934,2210,1478,741}; // -> 5KHz*/

/*{0,370,739,1105,1467,1823,2171,2511,2841,3160,3467,3760,4037,4299,4544,4772,4980,5168,5337,5484,
5609,5713,5794,5851,5886,5898,5886,5851,5794,5713,5609,5484,5337,5168,4980,4772,4544,4299,4037,376
0,3467,3160,2841,2511,2171,1823,1467,1105,739,370}; // -> 10KHz*/

{0,185,369,552,733,911,1085,1255,1420,1580,1733,1879,2018,2149,2271,2385,2489,2583,2667,2741,2804,
2855,2896,2925,2942,2948,2942,2925,2896,2855,2804,2741,2667,2583,2489,2385,2271,2149,2018,1879,173
3,1580,1420,1255,1085,911,733,552,369,185}; // -> 20KHz*/

/*{0,64,128,192,254,316,377,436,493,548,601,652,700,746,788,828,864,896,926,951,973,991,1005,1015,
1021,1023,1021,1015,1005,991,973,951,926,896,864,828,788,746,700,652,601,548,493,436,377,316,254,1
92,128,64}; // -> 58KHz */

/*{0,46,92,138,183,227,271,313,354,394,432,469,503,536,566,595,621,644,665,683,699,712,722,729,734
,735,734,729,722,712,699,683,665,644,621,595,566,536,503,469,432,394,354,313,271,227,183,138,92,46
}; // -> 80KHz*/

/*{0,93,185,277,369,461,552,643,733,822,911,999,1085,1171,1255,1338,1420,1501,1580,1657,1733,1807,
1879,1950,2018,2085,2149,2211,2271,2329,2385,2438,2489,2537,2583,2627,2667,2706,2741,2774,2804,283
1,2855,2877,2896,2912,2925,2935,2942,2947,2948,2947,2942,2935,2925,2912,2896,2877,2855,2831,2804,2
774,2741,2706,2667,2627,2583,2537,2489,2438,2385,2329,2271,2211,2149,2085,2018,1950,1879,1807,1733
,1657,1580,1501,1420,1338,1255,1171,1085,999,911,822,733,643,552,461,369,277,185,93}; // -> 20KHz
100 pontos*/

//-----
//Rotina de atendimento à interrupção do timer1
void __attribute__((__interrupt__)) _T1Interrupt(void)
{
    if (phase>=50) //Se chegou ao fim da tabela de seno
    {
        if (arcada_positiva == TRUE) //Verifica em que arcada estávamos
        {
            arcada_positiva = FALSE; //Sinaliza a nova arcada
            phase = 0; //Reset da fase
            SetDCMCPWM(mosfet,0,0); //Desliga o PWM de Q1
            mosfet = 2; //Sinaliza Q2
            OverrideMCPWM (PWM_GEN_2L & PWM_POUT_1L & PWM_POUT1L_INACT); //Reconfiguração dos
pinos de PWM
        }
        else
    }
}
```

```

inversor_main.c

{
    arcada_positiva = TRUE;           //Ou sinaliza esta arcada
    phase = 0;                         //Reset da fase
    SetDCMCPWM(mosfet,0,0);           //Desliga o PWM de Q2
    mosfet = 1;                       //Sinaliza Q1
    OverrideMCPWM (PWM_GEN_1L & PWM_POUT_2L & PWM_POUT2L_INACT); //Reconfiguração dos
pinos de PWM
}

if (pwm_constante == FALSE)          //Se não foi solicitado o modo de PWM constante
    SetDCMCPWM(mosfet, (sin_table[phase]/60)*amplitude,0); //Atualiza o duty-cycle do PWM
else                                  //Caso contrário o modo PWM constante foi solicitado
    SetDCMCPWM(mosfet, (sin_table[25]/60)*amplitude,0); //Atualiza o duty-cycle do PWM
phase++;                             //Incrementa a fase
IFS0bits.T1IF = 0;                   //Reset da flag de interrupção do timer1
}

//-----
//Rotina de atendimento à interrupção da ADC
void __attribute__((__interrupt__)) _ADCInterrupt(void)
{
    unsigned int aux;
    unsigned char i;
    ADCON1bits.ASAM = 0;              //Desliga o auto-sample da ADC
    WriteUART2('s');                  //Iníciol da trama
    WriteUART2('t');                  //Início2 da trama
    while(BusyUART2());
    for (i=0;i<6;i++)                 //Ler as primeiras 5 posições do buffer da ADC
    {
        aux = ReadADC10(i);
        WriteUART2((unsigned char)(aux >> 8)); //Envia o byte mais significativo
        WriteUART2((unsigned char) aux);        //Envia o byte menos significativo
        while(BusyUART2());
    }
    WriteUART2((unsigned char)0);       //Envia um dummy byte
    while(BusyUART2());
    IFS0bits.ADIF = 0;                  //Reset da flag de interrupção da ADC
}

//-----

void main (void)
{
    unsigned char aux;
    cfg_UART2();                       //Inicialização da UART2
    cfg_PWM ();                       //Inicialização do módulo de PWM
    cfg_T1();                         //Inicialização do timer 1
    cfg_ADC10();                     //Inicialização da ADC
    while(1)
    {
        if (DataRdyUART2())
        {
            aux = ReadUART2(); //Leitura de um caracter de comando

            switch (aux) //Identificação do comando
            {
                case 'a':
                    phase = 0; //Garante que a fase começa em 0
                    ConfigIntTimer1(T1_INT_PRIOR_7 & T1_INT_ON); // Liga o inversor
                    break;
                case 'b':
                    ConfigIntTimer1(T1_INT_OFF); //Desliga o inversor
                    SetDCMCPWM(mosfet,0,0); //Força um duty-cycle de 0
                    amplitude = 0; //Reset da amplitude
                    break;
                case '+':
                    if (amplitude < 60) amplitude++; //Aumento da amplitude
                    else putsUART2((unsigned int *)"Factor maximo de 60 atingido\r\n");
                    break;
                case '-':
                    if (amplitude > 0) amplitude--; //Diminuição da amplitude
                    else putsUART2((unsigned int *)"Factor minimo de 0 atingido\r\n");
                    break;
            }
        }
    }
}

```

```

                                                                    inversor_main.c

case 'c':
    if (pwm_constante == FALSE) pwm_constante = TRUE;
    else pwm_constante = FALSE;      //Liga ou desliga o modo SPWM
    break;
case 'd':
    ADCON1bits.ASAM = 1;      //Arranca com o auto-sample da ADC
    break;
case 'z':
    putsUART2((unsigned int *) "Msg teste para HyperTerminal\r\n"); //Para testar
a comunicação
    }
}
}

```

```

#include <p30fxxxx.h>
#include <adc10.h>
#include <uart.h>
#include <pwm.h>
#include <timer.h>

#define BRG (7372800*16)/(64*115200)-1 /* UxBRG=(OSC*PLL)/(64*BaudRate)-1
                                     Baudrate generator value for UART2 */

#define PWM_PERIOD 0x05C1 /* 0x0732 para 1KHz prescaler de 16,
                           0x1709 para 5KHz, 0x0B84 para 10KHz,
                           0x05C1 para 20k, 0x01FF para 58KHz,
                           // 0x016F para 80KHz */
#define T1_PERIOD 5898 // 5898 para 50 pontos, 2949 para 100 pontos

//-----
//Configura o módulo UART2 sem activar as interrupções
//Esta função usa rotinas definidas na "MPLAB C30 16-bit Peripheral Libraries" para dsPIC
void cfg_UART2 (void)
{
    unsigned int conf1,conf2;
    CloseUART2();

    conf1 = UART_EN & //liga a UART2
            UART_IDLE_CON & //em modo IDLE a UART2 continua a trabalhar
            UART_RX_TX & //comunicação da UART2 sem alternância dos pinos TX/RX
            UART_DIS_WAKE & //Wake-up on Start desligado
            UART_DIS_LOOPBACK & //Loopback desligado
            UART_DIS_ABAUD & //Função Auto-Baud desligada
            UART_NO_PAR_8BIT & //Modo 8 bits sem paridade
            UART_1STOPBIT; //Um stop-bit

    conf2 = UART_INT_TX & //Modo de interrupção para transmitir. Neste caso é
irrelevante
            UART_TX_PIN_NORMAL & //Transmit break pin em modo normal
            UART_TX_ENABLE & //Modo TX ligado
            UART_INT_RX_BUF_FUL & //Modo de interrupção para receber. Neste caso é
irrelevante
            UART_ADR_DETECT_DIS & //Address Detect desligado
            UART_RX_OVERRUN_CLEAR; //UART2 Overrun Bit Clear

    OpenUART2(conf1,conf2,BRG); //Configura os registos de controlo da UART2
}
//-----
//Configura o módulo ADC10 com interrupções
//Esta função usa rotinas definidas na "MPLAB C30 16-bit Peripheral Libraries" para dsPIC
void cfg_ADC10 (void)
{
    unsigned int channel,config1,config2,config3,port,scan;

    ADCON1bits.ADON = 0; //Garante que a ADC está desligada
    channel = ADC_CH0_NEG_SAMPLEA_NVREF; //Configura apenas a entrada - do canal S/H CH0

    config1 = ADC_MODULE_ON & //ADC ligada
              ADC_IDLE_CONTINUE & //Em modo IDLE mode o ADC continua a trabalhar
              ADC_FORMAT_INTG & //Formato da conversão
              ADC_CLK_AUTO & //Conversion trigger source
              ADC_AUTO_SAMPLING_OFF; //Auto sampling desligado.
              //Para arrancar com a ADC ligar este bit.

    config2 = ADC_VREF_EXT_AVSS & //Vref+ = AN0 (4.096V) e Vref- = AVSS (analog gnd)
              ADC_SCAN_ON & //CH0 faz o scan das entradas seleccionadas
              ADC_CONVERT_CH0 & //Converte apenas CH0
              ADC_SAMPLES_PER_INT_5 & //Gera uma interrupção ao fim de 5 samples
              ADC_ALT_BUF_OFF & //Buffer 1x16
              ADC_ALT_INPUT_OFF; //Apenas é usado o MUX A

    config3 = ADC_SAMPLE_TIME_31 & //Sample time = 31 x Tad
              ADC_CONV_CLK_SYSTEM & //ADC clock source = system clock (Crystal x PLL)/4
              ADC_CONV_CLK_8Tcy; //Conversion time = 8 x Tcy (271.3 ns)

    port = ENABLE_AN1_ANA & //Entrada AN1 -> VIN_SENSE
           ENABLE_AN2_ANA & //Entrada AN2 -> I_SENSE
           ENABLE_AN3_ANA & //Entrada AN3 -> Q1I_SENSE
           ENABLE_AN4_ANA & //Entrada AN4 -> Q2I_SENSE
           ENABLE_AN5_ANA; //Entrada AN5 -> TEMP

```

```

scan = SCAN_ALL &
      SKIP_SCAN_AN0 &
      SKIP_SCAN_AN6 &
      SKIP_SCAN_AN7 &
      SKIP_SCAN_AN8 &
      SKIP_SCAN_AN9 &           //O scan ocorre apenas para AN1, AN2, AN3, AN4 e AN5
      SKIP_SCAN_AN10 &
      SKIP_SCAN_AN11 &
      SKIP_SCAN_AN12 &
      SKIP_SCAN_AN13 &
      SKIP_SCAN_AN14 &
      SKIP_SCAN_AN15;

SetChanADC10(channel);           //Configura os canais S/H
OpenADC10(config1,config2,config3,port,scan); //Configura os registos de controlo da ADC
ConfigIntADC10(ADC_INT_ENABLE & ADC_INT_PRI_1); //Liga as interrupções com prioridade minima
}
//-----
//Configura o módulo PWM com interrupções
//Esta função usa rotinas definidas na "MPLAB C30 16-bit Peripheral Libraries" para dsPIC
void cfg_PWM (void)
{
    unsigned int sptime, config1, config2, config3, config;
    SetDCMCPWM(1,0,0);           //Garante que o duty_cycle register1 é 0
    SetDCMCPWM(2,0,0);           //Garante que o duty_cycle register2 é 0

    config = PWM_GEN_1L &        //PWM começa por controlar o pino PWM1L
             PWM_POUT_1H &
             PWM_POUT_2L &       //Estes pinos são descartados para já
             PWM_POUT_2H &
             PWM_POUT1L_INACT &
             PWM_POUT1H_INACT & //Força o estado Inactive quando os pinos
             PWM_POUT2L_INACT & //são descartados
             PWM_POUT2H_INACT;

    OverrideMCPWM(config);       //Configura o OVDCON register

    sptime = 0x0;                //SEVTSMP register não é usado
    config1 = PWM_EN &           //Liga o módulo de PWM
             PWM_IDLE_CON &      //Em modo IDLE mode o PWM continua a trabalhar
             PWM_OP_SCALE1 &     //PWM postscaler = 1
             PWM_IPCLK_SCALE1 & //PWM prescaler = 1
             PWM_MOD_FREE;       //Modo de funcionamento free
    config2 = PWM_MOD1_IND &      //Os pinos do PWM1 em modo independente
             PWM_PEN1L &         //Pino PWM1L ligado para Q1
             PWM_PDIS1H &        //Pino PWM1H desligado
             PWM_PEN2L &         //Pino PWM2L ligado para Q2
             PWM_PDIS2H;         //Pino PWM2H desligado

    config3 = PWM_SEVOPS1 &      //Special event posscaler = 1
             PWM_OSYNC_TCY &    //Sincronização feita pelo system clock
             PWM_UEN;           //PWM update ligado
    OpenMCPWM(PWM_PERIOD, sptime, config1, config2, config3); //Configura o módulo de PWM
}
//-----
//Configura o timer T1
//Esta função usa rotinas definidas na "MPLAB C30 16-bit Peripheral Libraries" para dsPIC
void cfg_T1 (void)
{
    unsigned int timer1config, timer1_period;
    timer1config = T1_ON &       //Liga o timer1
                 T1_IDLE_CON &   //Em modo IDLE mode o timer1 continua a trabalhar
                 T1_GATE_OFF &  //Gate time accumulation desligado
                 T1_PS_1_1 &    //Prescaler = 1
                 T1_SYNC_EXT_OFF & //Sincronização externa desligada
                 T1_SOURCE_INT;  //Timer clock source = system clock
    timer1_period = T1_PERIOD;   //Configura o periodo
    WriteTimer1(0);              //Reset do contador
    OpenTimer1(timer1config, timer1_period); //Configura o timer1
    ConfigIntTimer1(T1_INT_OFF); //Para já as interrupções estão desligadas
}
//-----

```

Anexo F – Código fonte da interface de controlo

1

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace teste
{
    //Variáveis Globais
    public partial class Inverter : Form
    {
        public char[] comandos = { 'a', 'b', '+', '-', 'd' };
        public byte[] buffer={0,0,0,0,0,0,0,0,0,0,0,0};
        public int[] resultado = {0,0,0,0,0,0};
        public int aux=0,amplitude=0;
        public Inverter()
        {
            InitializeComponent();

            //-----
            //Rotina da checkBox PWM, liga ou desliga o módulo
            private void checkBox1_CheckedChanged(object sender, EventArgs e)
            {
                if (turn_on_pwm.Checked == true)
                {
                    if (turn_on_rs232.Checked == true) port1.Write(comandos, 0, 1);
                }
                else
                {
                    if (turn_on_rs232.Checked == true) port1.Write(comandos, 1, 1);

                    //Reset da amplitude e respectiva escrita no label
                    amplitude = 0;
                    valor_amplitude.Text = Convert.ToString((amplitude * 100) / 60);
                    valor_amplitude.Text = string.Concat(valor_amplitude.Text, '%');
                    vertical_bar_amp.Value = (amplitude * 100) / 60;
                }
            }
            //-----
            //Rotina da checkBox RS-232, liga ou desliga a porta de comunicação
            private void checkBox2_CheckedChanged(object sender, EventArgs e)
            {
                if (turn_on_rs232.Checked == true) port1.Open();
                else port1.Close();
            }
            //-----
            //Rotina do timer interno
            //A cada 750ms manda desencadear leituras dos 5 canais da ADC (valores inteiros
            //e positivos de 10 bits)
            //Calcula o valor das grandezas e escreve-as nos respectivos labels
            private void timer1_Tick(object sender, EventArgs e)
            {
                if (turn_on_rs232.Checked == true)
                {
                    port1.Write(comandos, 4, 1);

                    //Tensão de entrada do gerador resultante do divisor resistivo R5 e R6
                    v_gerador.Text = Convert.ToString(Math.Round(resultado[0]*4*9.3/1000,1));
                    v_gerador.Text = string.Concat(v_gerador.Text, 'V');

                    //Corrente de saída medida à saída de IC4. O ganho total do sensor
                    //de corrente e amplificador é de 3.57, com um offset próximo de
                    //2.05V
                    i_saida.Text = Convert.ToString(Math.Round((resultado[1]-518)*4/3.57));
                    i_saida.Text = string.Concat(i_saida.Text, "mA");

                    //Correntes nos transistores medidas nas saídas de IC5
                    //Estas correntes atravessam R1IS e R2IS e são amplificadas por um factor
                    //de 51
                    i_Q1.Text = Convert.ToString(Math.Round(resultado[2]*4/(1000*0.51),1));
                }
            }
        }
    }
}
```



```

        i_Q1.Text = string.Concat(i_Q1.Text, 'A');
        i_Q2.Text = Convert.ToString(Math.Round(resultado[3]*4/(1000*0.51),1));
        i_Q2.Text = string.Concat(i_Q2.Text, 'A');

        //Leitura do sensor de temperatura. A sensibilidade do
        //sensor é de 10mV/°K
        temperatura.Text = Convert.ToString(resultado[4]*4/10-273);
        temperatura.Text = string.Concat(temperatura.Text, "°C");
    }

}

//-----
//Recepção e processamento dos dados recebidos (5 leituras da ADC)
private void port1_DataReceived(object sender, System.IO.Ports.
SerialDataReceivedEventArgs e)
{
    int i;
    port1.Read(buffer, 0, 1);
    if (buffer[0] == 's') //Identificação do 1º início de trama
    {
        port1.Read(buffer, 0, 1);
        if (buffer[0] == 't') //Identificação do 2º início de trama
        {
            port1.Read(buffer, 0, 11); //Leitura de 1 bloco de 11 bytes

            //Reconstrução dos dados em valores de 16 bits
            for (i = 0; i < 5; i++)
            {
                aux = (int)buffer[2 * i];
                aux = aux << 8;
                resultado[i] = (aux | (int)buffer[2 * i + 1]);
            }
        }
    }
}

//-----
//Botão de aumento da amplitude
private void up_Click(object sender, EventArgs e)
{
    if (amplitude < 60)
    {
        amplitude++; //Incremento do factor de amplitude
        valor_amplitude.Text = Convert.ToString((amplitude * 100) / 60);
        vertical_bar_amp.Value = (amplitude * 100) / 60;
        valor_amplitude.Text = string.Concat(valor_amplitude.Text, '%');
        if (turn_on_rs232.Checked == true) port1.Write(comandos, 2, 1);
    }
}

//Botão de diminuição da amplitude
private void down_Click(object sender, EventArgs e)
{
    if (amplitude > 0)
    {
        amplitude--; //Decremento do factor de amplitude
        valor_amplitude.Text = Convert.ToString((amplitude * 100) / 60);
        vertical_bar_amp.Value = (amplitude * 100) / 60;
        valor_amplitude.Text = string.Concat(valor_amplitude.Text, '%');
        if (turn_on_rs232.Checked == true) port1.Write(comandos, 3, 1);
    }
}
}
}

```

Anexo G – Documentos de apoio com as livrarias dos periféricos usados no dsPIC

dsPIC30F/dsPIC33F/PIC24H Timers Peripheral Module Library Help

Table of Contents

1	Library Features
2	Using the Library Functions in Your Code
3	Functions
3.1	<i>CloseTimerx</i>
3.2	<i>CloseTimerxy</i>
3.3	<i>ConfigIntTimerx</i>
3.4	<i>ConfigIntTimerxy</i>
3.5	<i>OpenTimerx</i>
3.6	<i>OpenTimerxy</i>
3.7	<i>ReadTimerx</i>
3.8	<i>ReadTimerxy</i>
3.9	<i>WriteTimerx</i>
3.10	<i>WriteTimerxy</i>
4	Macros
4.1	<i>EnableIntTx</i>
4.2	<i>DisableIntTx</i>
4.3	<i>SetPriorityIntTx</i>

1 Library Features

This peripheral library module provides:

- Module initialization functions.
- Simple functions to configure timers for different modes and 16/ 32 bit operation.
- Functions to read timer registers and configure interrupts.
- Simple interface macros to enable/disable interrupts.

2 Using the Library Functions in Your Code

Library routine parameters can be constructed using an AND based mask. For more information on this masks, see [16-bit Peripheral Libraries](#). An example of use is shown below.

dsPIC30F Example of Use (AND mask)

```
#define __dsPIC30F6014__
#include <p30fxxxx.h>
#include<timer.h>
unsigned int timer_value;
void __attribute__((__interrupt__)) _T1Interrupt(void)
{
    PORTDbits.RD1 = 1;    /* turn off LED on RD1 */
    WriteTimer1(0);
    IFS0bits.T1IF = 0;    /* Clear Timer interrupt flag */
}
int main(void)
{
    unsigned int match_value;
    TRISDbits.TRISD1 = 0;
    PORTDbits.RD1 = 1;    /* turn off LED on RD1 */
    /* Enable Timer1 Interrupt and Priority to "1" */
    ConfigIntTimer1(T1_INT_PRIOR_1 & T1_INT_ON);
    WriteTimer1(0);
    match_value = 0xFFFF;
    OpenTimer1(T1_ON & T1_GATE_OFF & T1_IDLE_STOP &
               T1_PS_1_1 & T1_SYNC_EXT_OFF &
               T1_SOURCE_INT, match_value);
    /* Wait till the timer matches with the period value */
    while(1)
    {
        timer_value = ReadTimer1();
        if(timer_value >= 0x7FFF)
        {
            PORTDbits.RD1 = 0; /* turn on LED on RD1 */
        }
    }
    CloseTimer1();
}
```

3 Functions

3.1 CloseTimerx

Function Prototype	<pre>void CloseTimer1(void); void CloseTimer2(void); void CloseTimer3(void); void CloseTimer4(void); void CloseTimer5(void); void CloseTimer6(void); void CloseTimer7(void); void CloseTimer8(void);</pre>
---------------------------	--

Include	<code>void CloseTimer9(void);</code> <code>timer.h</code>
Description	This function turns off the 16-bit timer module.
Arguments	None
Return Value	None
Remarks:	This function first disables the 16-bit timer interrupt and then turns off the timer module. The Interrupt Flag bit (TxIF) is also cleared.
Source File:	<code>CloseTimer1.c</code> <code>CloseTimer2.c</code> <code>CloseTimer3.c</code> <code>CloseTimer4.c</code> <code>CloseTimer5.c</code> <code>CloseTimer6.c</code> <code>CloseTimer7.c</code> <code>CloseTimer8.c</code> <code>CloseTimer9.c</code>
Code Example	<code>CloseTimer1();</code>

3.2 CloseTimerxy

Function Prototype	<code>void CloseTimer23(void);</code> <code>void CloseTimer45(void);</code> <code>void CloseTimer67(void);</code> <code>void CloseTimer89(void);</code>
Include	<code>timer.h</code>
Description	This function turns off the 32-bit timer module.
Arguments	None
Return Value	None
Remarks:	This function disables the 32-bit timer interrupt and then turns off the timer module. The Interrupt Flag bit (TxIF) is also cleared. CloseTimer23 turns off Timer2 and disables Timer3 Interrupt. CloseTimer45 turns off Timer4 and disables Timer5 Interrupt.
Source File:	<code>CloseTimer23.c</code> <code>CloseTimer45.c</code> <code>CloseTimer67.c</code> <code>CloseTimer89.c</code>
Code Example	<code>CloseTimer23();</code>

3.3 ConfigIntTimerx

Function Prototype	<code>void ConfigIntTimer1(unsigned int config);</code> <code>void ConfigIntTimer2(unsigned int config);</code> <code>void ConfigIntTimer3(unsigned int config);</code> <code>void ConfigIntTimer4(unsigned int config);</code> <code>void ConfigIntTimer5(unsigned int config);</code> <code>void ConfigIntTimer6(unsigned int config);</code> <code>void ConfigIntTimer7(unsigned int config);</code> <code>void ConfigIntTimer8(unsigned int config);</code> <code>void ConfigIntTimer9(unsigned int config);</code>
Include	<code>timer.h</code>
Description	This function configures the 16-bit timer interrupt.
Arguments	<i>config</i> - Timer interrupt priority and enable/disable information as defined below: Timer Interrupt Priority Tx_INT_PRIOR_7 Tx_INT_PRIOR_6 Tx_INT_PRIOR_5 Tx_INT_PRIOR_4 Tx_INT_PRIOR_3 Tx_INT_PRIOR_2 Tx_INT_PRIOR_1 Tx_INT_PRIOR_0

	Timer Interrupt Enable Tx_INT_ON Tx_INT_OFF
Return Value	None.
Remarks:	This function clears the 16-bit Interrupt Flag (TxIF) bit and then sets the interrupt priority and enables/disables the interrupt.
Source File:	ConfigIntTimer1.c ConfigIntTimer2.c ConfigIntTimer3.c ConfigIntTimer4.c ConfigIntTimer5.c ConfigIntTimer6.c ConfigIntTimer7.c ConfigIntTimer8.c ConfigIntTimer9.c
Code Example	ConfigIntTimer1(T1_INT_PRIOR_3 & T1_INT_ON);

3.4 ConfigIntTimerxy

Function Prototype	void ConfigIntTimer23(unsigned int <i>config</i>); void ConfigIntTimer45(unsigned int <i>config</i>); void ConfigIntTimer67(unsigned int <i>config</i>); void ConfigIntTimer89(unsigned int <i>config</i>);
Include	timer.h
Description	This function configures the 32-bit timer interrupt.
Arguments	<i>config</i> - Timer interrupt priority and enable/disable information as defined below: Timer Interrupt Priority Tx_INT_PRIOR_7 Tx_INT_PRIOR_6 Tx_INT_PRIOR_5 Tx_INT_PRIOR_4 Tx_INT_PRIOR_3 Tx_INT_PRIOR_2 Tx_INT_PRIOR_1 Tx_INT_PRIOR_0 Timer Interrupt Enable Tx_INT_ON Tx_INT_OFF
Return Value	None
Remarks:	This function clears the 32-bit Interrupt Flag (TxIF) bit and then sets the interrupt priority and enables/disables the interrupt.
Source File:	ConfigIntTimer23.c ConfigIntTimer45.c ConfigIntTimer67.c ConfigIntTimer89.c
Code Example	ConfigIntTimer23(T3_INT_PRIOR_5 & T3_INT_ON);

3.5 OpenTimerx

Function Prototype	void OpenTimer1(unsigned int <i>config</i> , unsigned int <i>period</i>); void OpenTimer2(unsigned int <i>config</i> , unsigned int <i>period</i>); void OpenTimer3(unsigned int <i>config</i> , unsigned int <i>period</i>); void OpenTimer4(unsigned int <i>config</i> , unsigned int <i>period</i>); void OpenTimer5(unsigned int <i>config</i> , unsigned int <i>period</i>); void OpenTimer6(unsigned int <i>config</i> , unsigned int <i>period</i>); void OpenTimer7(unsigned int <i>config</i> ,
---------------------------	--

	<pre> unsigned int <i>period</i>); void OpenTimer8(unsigned int <i>config</i>, unsigned int <i>period</i>); void OpenTimer9(unsigned int <i>config</i>, unsigned int <i>period</i>); </pre>
Include	timer.h
Description	This function configures the 16-bit timer module.
Arguments	<p><i>config</i> - This contains the parameters to be configured in the TxCON register as defined below:</p> <p>Timer Module On/Off</p> <pre> Tx_ON Tx_OFF </pre> <p>Timer Module Idle mode On/Off</p> <pre> Tx_IDLE_CON Tx_IDLE_STOP </pre> <p>Timer Gate time accumulation enable</p> <pre> Tx_GATE_ON Tx_GATE_OFF </pre> <p>Timer prescaler</p> <pre> Tx_PS_1_1 Tx_PS_1_8 Tx_PS_1_64 Tx_PS_1_128 </pre> <p>Timer Synchronous clock enable</p> <pre> Tx_SYNC_EXT_ON Tx_SYNC_EXT_OFF </pre> <p>Timer clock source</p> <pre> Tx_SOURCE_EXT Tx_SOURCE_INT </pre>
	<i>period</i> - This contains the period match value to be stored into the PR register.
Return Value	None
Remarks:	This function configures the 16-bit Timer Control register and sets the period match value into the PR register.
Source File:	<pre> OpenTimer1.c OpenTimer2.c OpenTimer3.c OpenTimer4.c OpenTimer5.c OpenTimer6.c OpenTimer7.c OpenTimer8.c OpenTimer9.c </pre>
Code Example	<pre> OpenTimer1(T1_ON & T1_GATE_OFF & T1_PS_1_8 & T1_SYNC_EXT_OFF & T1_SOURCE_INT, 0xFF); </pre>

3.6 OpenTimerxy

Function Prototype	<pre> void OpenTimer23(unsigned int <i>config</i>, unsigned long <i>period</i>); void OpenTimer45(unsigned int <i>config</i>, unsigned long <i>period</i>); void OpenTimer67(unsigned int <i>config</i>, unsigned long <i>period</i>); void OpenTimer89(unsigned int <i>config</i>, unsigned long <i>period</i>); </pre>
Include	timer.h
Description	This function configures the 32-bit timer module.
Arguments	<i>config</i> - This contains the parameters to be configured in the TxCON register as defined below:

Timer module On/Off

Tx_ON
Tx_OFF

Timer Module Idle mode On/Off

Tx_IDLE_CON
Tx_IDLE_STOP

Timer Gate time accumulation enable

Tx_GATE_ON
Tx_GATE_OFF

Timer prescaler

Tx_PS_1_1
Tx_PS_1_8
Tx_PS_1_64
Tx_PS_1_128

Timer Synchronous clock enable

Tx_SYNC_EXT_ON
Tx_SYNC_EXT_OFF

Timer clock source

Tx_SOURCE_EXT
Tx_SOURCE_INT

period - This contains the period match value to be stored into the 32-bit PR register.

Return Value

None

Remarks:

This function configures the 32-bit Timer Control register and sets the period match value into the PR register.

Source File:

OpenTimer23.c
OpenTimer45.c
OpenTimer67.c
OpenTimer89.c

Code Example

```
OpenTimer23(T2_ON & T2_GATE_OFF & T2_PS_1_8 &  
T2_32BIT_MODE_ON & T2_SYNC_EXT_OFF & T2_SOURCE_INT,  
0xFFFF);
```

3.7 ReadTimerx

Function Prototype

```
unsigned int ReadTimer1(void);  
unsigned int ReadTimer2(void);  
unsigned int ReadTimer3(void);  
unsigned int ReadTimer4(void);  
unsigned int ReadTimer5(void);  
unsigned int ReadTimer6(void);  
unsigned int ReadTimer7(void);  
unsigned int ReadTimer8(void);  
unsigned int ReadTimer9(void);  
timer.h
```

Include

Description

This function reads the contents of the 16-bit Timer register.

Arguments

None.

Return Value

16 bit contents of the TMR register

Remarks:

This function returns the contents of the 16-bit TMR register.

Source File:

ReadTimer1.c
ReadTimer2.c
ReadTimer3.c
ReadTimer4.c
ReadTimer5.c
ReadTimer6.c
ReadTimer7.c
ReadTimer8.c
ReadTimer9.c

Code Example

```
unsigned int timer1_value;  
timer1_value = ReadTimer1();
```

3.8 *ReadTimerxy*

Function Prototype unsigned long ReadTimer23(void);
 unsigned long ReadTimer45(void);
 unsigned long ReadTimer67(void);
 unsigned long ReadTimer89(void);

Include timer.h

Description This function reads the contents of the 32-bit Timer register.

Arguments None

Return Value 32-bit contents of the TMR register

Remarks: This function returns the contents of the 32-bit TMR register

Source File: ReadTimer23.c
 ReadTimer45.c
 ReadTimer67.c
 ReadTimer89.c

Code Example unsigned long timer23_value;
 timer23_value = ReadTimer23();

3.9 *WriteTimerx*

Function Prototype void WriteTimer1(unsigned int timer);
 void WriteTimer2(unsigned int timer);
 void WriteTimer3(unsigned int timer);
 void WriteTimer4(unsigned int timer);
 void WriteTimer5(unsigned int timer);
 void WriteTimer6(unsigned int timer);
 void WriteTimer7(unsigned int timer);
 void WriteTimer8(unsigned int timer);
 void WriteTimer9(unsigned int timer);

Include timer.h

Description This function writes the 16-bit value into the Timer register.

Arguments *timer* - This is the 16-bit value to be stored into TMR register.

Return Value None.

Remarks: None

Source File: WriteTimer1.c
 WriteTimer2.c
 WriteTimer3.c
 WriteTimer4.c
 WriteTimer5.c
 WriteTimer6.c
 WriteTimer7.c
 WriteTimer8.c
 WriteTimer9.c

Code Example unsigned int timer_init = 0xAB;
 WriteTimer1(timer_init);

3.10 *WriteTimerxy*

Function Prototype void WriteTimer23(unsigned long timer);
 void WriteTimer45(unsigned long timer);
 void WriteTimer67(unsigned long timer);
 void WriteTimer89(unsigned long timer);

Include timer.h

Description This function writes the 32-bit value into the Timer register.

Arguments *timer* - This is the 32-bit value to be stored into TMR register.

Return Value None.

Remarks: None

Source File: WriteTimer23.c
 WriteTimer45.c
 WriteTimer67.c
 WriteTimer89.c

Code Example unsigned long timer23_init = 0xABCD;


```
WriteTimer23(timer23_init);
```

4 Macros

4.1 *EnableIntTx*

Macro	EnableIntT1 EnableIntT2 EnableIntT3 EnableIntT4 EnableIntT5
Overview	This macro sets Timer Interrupt Enable bit of Interrupt Enable Control register.
Input	None
Output	None
Remarks	None
Code Example	EnableIntT1;

4.2 *DisableIntTx*

Macro	DisableIntT1 DisableIntT2 DisableIntT3 DisableIntT4 DisableIntT5
Overview	This macro clears Timer Interrupt Enable bit of Interrupt Enable Control register.
Input	None
Output	None
Remarks	None
Code Example	DisableIntT2;

4.3 *SetPriorityIntTx*

Macro	SetPriorityIntT1 SetPriorityIntT2 SetPriorityIntT3 SetPriorityIntT4 SetPriorityIntT5
Overview	This macro sets Timer Interrupt Priority bits of Interrupt Priority Control register.
Input	Interrupt Priority Level.
Output	None
Remarks	None
Code Example	SetPriorityIntT4(7);

dsPIC30F/dsPIC33F/PIC24H

UART Peripheral Module Library Help

Table of Contents

- 1 Library Features**
- 2 Using the Library Functions in Your Code**
- 3 Functions**
 - 3.1 *BusyUARTx*
 - 3.2 *CloseUARTx*
 - 3.3 *ConfigIntUARTx*
 - 3.4 *DataRdyUARTx*
 - 3.5 *getsUARTx*
 - 3.6 *OpenUARTx [dsPIC30F]*
 - 3.7 *OpenUARTx [dsPIC33F/PIC24H]*
 - 3.8 *putsUARTx*
 - 3.9 *ReadUARTx or getcUARTx*
 - 3.10 *WriteUARTx or putcUARTx*
- 4 Macros**
 - 4.1 *EnableIntUxRX*
 - 4.2 *EnableIntUxTX*
 - 4.3 *DisableIntUxRX*
 - 4.4 *DisableIntUxTX*
 - 4.5 *SetPriorityIntUxRX*
 - 4.6 *SetPriorityIntUxTX*

1 Library Features

This peripheral library module provides:

- Module initialization functions.
- Simple functions to read from and write to the buffers, supporting both 8-bit and 9-bit transmission and reception.
- Multiple byte/word transmission or reception in a single function call.
- Simple interface macros to enable/disable interrupts.

2 Using the Library Functions in Your Code

Library routine parameters can be constructed using an AND based mask. For more information on this masks, see [16-bit Peripheral Libraries](#). An example of use is shown below.

dsPIC30F Example of Use (AND mask)

```
#define __dsPIC30F6014__
#include<p30fxxx.h>
#include<uart.h>
/* Received data is stored in array Buf */
char Buf[80];
char * Receiveddata = Buf;
/* This is UART1 transmit ISR */
void __attribute__((__interrupt__)) _U1TXInterrupt(void)
{
    IFS0bits.U1TXIF = 0;
}
/* This is UART1 receive ISR */
void __attribute__((__interrupt__)) _U1RXInterrupt(void)
{
    IFS0bits.U1RXIF = 0;
    /* Read the receive buffer till atleast one or more character can be read */
    while( DataRdyUART1())
    {
        ( *( Receiveddata)++) = ReadUART1();
    }
}
int main(void)
{
    /* Data to be transmitted using UART communication module */
    char Txdata[] = {'M','i','c','r','o','c','h','i','p',' ','I','C','D','2','\0'};
    /* Holds the value of baud register */
    unsigned int baudvalue;
    /* Holds the value of uart config reg */
    unsigned int U1MODEvalue;
    /* Holds the information regarding uart TX & RX interrupt modes */
    unsigned int U1STAValue;
    /* Turn off UART1module */
    CloseUART1();
    /* Configure uart1 receive and transmit interrupt */
    ConfigIntUART1(UART_RX_INT_EN & UART_RX_INT_PR6 &
        UART_TX_INT_DIS & UART_TX_INT_PR2);
    /* Configure UART1 module to transmit 8 bit data with one stopbit. Also Enable loopback mode */
    baudvalue = 5;
    U1MODEvalue = UART_EN & UART_IDLE_CON &
        UART_DIS_WAKE & UART_EN_LOOPBACK &
        UART_EN_ABAUD & UART_NO_PAR_8BIT &
```

```

        UART_1STOPBIT;
    U1STAValue = UART_INT_TX_BUF_EMPTY &
        UART_TX_PIN_NORMAL &
        UART_TX_ENABLE & UART_INT_RX_3_4_FUL &
        UART_ADR_DETECT_DIS &
        UART_RX_OVERRUN_CLEAR;
    OpenUART1(U1MODEValue, U1STAValue, baudvalue);

    /* Load transmit buffer and transmit the same till null character is
    encountered */
    putsUART1 ((unsigned int *)Txdata);
    /* Wait for transmission to complete */
    while(BusyUART1());
    /* Read all the data remaining in receive buffer which are unread */
    while(DataRdyUART1())
    {
        (*( Receiveddata)++) = ReadUART1() ;
    }
    /* Turn off UART1 module */
    CloseUART1();
    return 0;
}

```

3 Functions

3.1 BusyUARTx

Function Prototype	char BusyUART1(void); char BusyUART2(void);
Include	uart.h
Description	This function returns the UART transmission status.
Arguments	None
Return Value	If '1' is returned, it indicates that UART is busy in transmission and UxSTA<TRMT> bit is '0'. If '0' is returned, it indicates that UART is not busy and UxSTA<TRMT> bit is '1'.
Remarks:	This function returns the status of the UART. This indicates if the UART is busy in transmission as indicated by the UxSTA<TRMT> bit.
Source File:	BusyUART1.c BusyUART2.c
Code Example	while(BusyUART1());

3.2 CloseUARTx

Function Prototype	void CloseUART1(void); void CloseUART2(void);
Include	uart.h
Description	This function turns off the UART module.
Arguments	None
Return Value	None
Remarks:	This function first turns off the UART module and then disables the UART transmit and receive interrupts. The Interrupt Flag bits are also cleared.
Source File:	CloseUART1.c CloseUART2.c
Code Example	CloseUART1();

3.3 ConfigIntUARTx

Function Prototype	void ConfigIntUART1(unsigned int config); void ConfigIntUART2(unsigned int config);
Include	uart.h

Description	This function configures the UART Interrupts.
Arguments	<i>config</i> - Individual interrupt enable/disable information as defined below. Receive Interrupt enable UART_RX_INT_EN UART_RX_INT_DIS Receive Interrupt Priority UART_RX_INT_PR0 UART_RX_INT_PR1 UART_RX_INT_PR2 UART_RX_INT_PR3 UART_RX_INT_PR4 UART_RX_INT_PR5 UART_RX_INT_PR6 UART_RX_INT_PR7 Transmit Interrupt Enable UART_TX_INT_EN UART_TX_INT_DIS Transmit Interrupt Priority UART_TX_INT_PR0 UART_TX_INT_PR1 UART_TX_INT_PR2 UART_TX_INT_PR3 UART_TX_INT_PR4 UART_TX_INT_PR5 UART_TX_INT_PR6 UART_TX_INT_PR7
Return Value	None
Remarks:	This function enables/disables the UART transmit and receive interrupts and sets the interrupt priorities.
Source File:	ConfigIntUART1.c ConfigIntUART2.c
Code Example	ConfigIntUART1 (UART_RX_INT_EN & UART_RX_INT_PR5 & UART_TX_INT_EN & UART_TX_INT_PR3);

3.4 DataRdyUARTx

Function Prototype	char DataRdyUART1(void); char DataRdyUART2(void);
Include	uart.h
Description	This function returns the UART receive buffer status.
Arguments	None
Return Value	If '1' is returned, it indicates that the receive buffer has a data to be read. If '0' is returned, it indicates that receive buffer does not have any new data to be read.
Remarks:	This function returns the status of the UART receive buffer. This indicates if the UART receive buffer contains any new data that is yet to be read as indicated by the UxSTA<URXDA> bit.
Source File:	DataRdyUART1.c DataRdyUART2.c
Code Example	while(DataRdyUART1());

3.5 getsUARTx

Function Prototype	unsigned int getsUART1(unsigned int <i>length</i> , unsigned int * <i>buffer</i> , unsigned int <i>uart_data_wait</i>); unsigned int getsUART2(unsigned int <i>length</i> , unsigned int * <i>buffer</i> , unsigned int <i>uart_data_wait</i>);
Include	uart.h
Description	This function reads a string of data of specified length and stores it into the buffer location specified.
Arguments	<i>length</i> - This is the length of the string to be received.

	<i>buffer</i> - This is the pointer to the location where the data received have to be stored.
	<i>uart_data_wait</i> - This is the time-out count for which the module has to wait before return. If the time-out count is 'N', the actual time out would be about (19 * N – 1) instruction cycles.
Return Value	This function returns the number of bytes yet to be received. If the return value is '0', it indicates that the complete string has been received. If the return value is non-zero, it indicates that the complete string has not been received.
Remarks:	None.
Source File:	getsSPI1.c getsSPI2.c
Code Example	Datarem = getsSPI1(6, Rxdata_loc, 40);

3.6 OpenUARTx [dsPIC30F]

Function Prototype	<pre>void OpenUART1(unsigned int config1, unsigned int config2, unsigned int ubrg); void OpenUART2(unsigned int config1, unsigned int config2, unsigned int ubrg);</pre>
Include	uart.h
Description	This function configures the UART module.
Arguments	<p><i>config1</i> - This contains the parameters to be configured in the UxMODE register as defined below.</p> <p>UART Enable/Disable UART_EN UART_DIS</p> <p>UART Idle mode operation UART_IDLE_CON UART_IDLE_STOP</p> <p>UART communication with alternate TX/RX pins* UART_ALTRX_ALTTX UART_RX_TX</p> <p>* UART communication with alternate pins is available only for certain devices and the suitable data sheet should be referred to.</p> <p>UART Wake-up on Start UART_EN_WAKE UART_DIS_WAKE</p> <p>UART Loopback mode enable/disable UART_EN_LOOPBACK UART_DIS_LOOPBACK</p> <p>Input to Capture module UART_EN_ABAUD UART_DIS_ABAUD</p> <p>Parity and data bits select UART_NO_PAR_9BIT UART_ODD_PAR_8BIT UART_EVEN_PAR_8BIT UART_NO_PAR_8BIT</p> <p>Number of Stop bits UART_2STOPBITS UART_1STOPBIT</p> <p><i>config2</i> - This contains the parameters to be configured in the UxSTA register as defined below.</p> <p>UART Transmission Interrupt Mode select UART_INT_TX_BUF_EMPTY UART_INT_TX</p>

UART Transmit Break bit

UART_TX_PIN_NORMAL

UART_TX_PIN_LOW

UART Transmit Enable/Disable

UART_TX_ENABLE

UART_TX_DISABLE

UART Receive Interrupt Mode select

UART_INT_RX_BUF_FUL

UART_INT_RX_3_4_FUL

UART_INT_RX_CHAR

UART Address Detect Enable/Disable

UART_ADR_DETECT_EN

UART_ADR_DETECT_DIS

UART Overrun Bit Clear

UART_RX_OVERRUN_CLEAR

ubrg - This is the value to be written into UxBRG register to set the baud rate.

Return Value

None

Remarks:

This functions configures the UART transmit and receive sections and sets the communication baud rate.

Source File:

OpenUART1.c

OpenUART2.c

Code Example

baud = 5;

```
UMODEvalue = UART_EN & UART_IDLE_CON &
UART_DIS_WAKE & UART_EN_LOOPBACK &
UART_EN_ABAUD & UART_NO_PAR_8BIT &
UART_1STOPBIT;
```

```
U1STAv alue = UART_INT_TX_BUF_EMPTY &
UART_TX_PIN_NORMAL &
UART_TX_ENABLE &
UART_INT_RX_3_4_FUL &
UART_ADR_DETECT_DIS &
UART_RX_OVERRUN_CLEAR;
```

```
OpenUART1 (U1MODEvalue, U1STAv alue, baud);
```

3.7 OpenUARTx [dsPIC33F/PIC24H]

Function Prototype

```
void OpenUART1(unsigned int config1,
unsigned int config2, unsigned int ubrg);
void OpenUART2(unsigned int config1,
unsigned int config2, unsigned int ubrg);
uart.h
```

Include

Description

This function configures the UART module.

Arguments

config1 - This contains the parameters to be configured in the UxMODE register as defined below.

UART Enable/Disable

UART_EN

UART_DIS

UART Idle mode operation

UART_IDLE_CON

UART_IDLE_STOP

IrDA Encoding/Decoding Enable/Disable

UART_IrDA_ENABLE

UART_IrDA_DISABLE

UxRTS Pin Mode Select

UART_MODE_SIMPLEX

UART_MODE_FLOW

UxCTS and UxRTS Pin Usage Mode*

UART_UEN_11

```

UART_UEN_10
UART_UEN_01
UART_UEN_00

```

* Refer to the data sheet for specific details on these 4 modes.

UART communication with alternate TX/RX pins**

```

UART_ALTRX_ALTTX
UART_RX_TX

```

** UART communication with alternate pins is available only for certain devices and the suitable data sheet should be referred to.

UART Wake-up on Start

```

UART_EN_WAKE
UART_DIS_WAKE

```

UART Loopback Mode Enable/Disable

```

UART_EN_LOOPBACK
UART_DIS_LOOPBACK

```

Autobaud Measurement Enable/Disable

```

UART_EN_ABAUD
UART_DIS_ABAUD

```

Parity and data bits select

```

UART_NO_PAR_9BIT
UART_ODD_PAR_8BIT
UART_EVEN_PAR_8BIT
UART_NO_PAR_8BIT

```

High Baud Rate Enable/Disable

```

UART_BRGH_FOUR
UART_BRGH_SIXTEEN

```

Number of Stop bits

```

UART_2STOPBITS
UART_1STOPBIT

```

config2 - This contains the parameters to be configured in the UxSTA register as defined below.

UART Transmission Interrupt Mode select

```

UART_INT_TX_BUF_EMPTY
UART_INT_TX

```

IrDA Polarity Inversion select

```

UART_IrDA_POL_INV_ONE
UART_IrDA_POL_INV_ZERO

```

UART Sync Break Character Transmission Enable/Disable

```

UART_SYNC_BREAK_ENABLED
UART_SYNC_BREAK_DISABLED

```

UART Transmit Enable/Disable

```

UART_TX_ENABLE
UART_TX_DISABLE

```

UART Receive Interrupt Mode select

```

UART_INT_RX_BUF_FUL
UART_INT_RX_3_4_FUL
UART_INT_RX_CHAR

```

UART Address Detect Enable/Disable

```

UART_ADR_DETECT_EN
UART_ADR_DETECT_DIS

```

UART Overrun Bit Clear

```

UART_RX_OVERRUN_CLEAR

```

ubrg - This is the value to be written into UxBRG register to set the baud rate.

Return Value

None

Remarks:

This function configures the UART transmit and receive sections and sets the communication baud rate.

Source File:

OpenUART1.c
OpenUART2.c

Code Example

baud = 5;


```

UMODEvalue = UART_EN & UART_IDLE_CON &
UART_IrDA_DISABLE & UART_MODE_SIMPLEX &
UART_UEN_00 & UART_DIS_WAKE &
UART_EN_LOOPBACK & UART_EN_ABAUD &
UART_BRGH_SIXTEEN & UART_NO_PAR_8BIT & UART_1STOPBIT;

U1STAvale = UART_INT_TX_BUF_EMPTY &
UART_TX_PIN_NORMAL & UART_TX_ENABLE &
UART_INT_RX_3_4_FUL & UART_ADR_DETECT_DIS &
UART_RX_OVERRUN_CLEAR & UART_IrDA_POL_INV_ZERO &
UART_SYNC_BREAK_DISABLED;

OpenUART1 (U1MODEvalue, U1STAvale, baud);

```

3.8 putsUARTx

Function Prototype	void putsUART1(unsigned int *buffer); void putsUART2(unsigned int *buffer);
Include	uart.h
Description	This function writes a string of data to be transmitted into the UART transmit buffer.
Arguments	<i>buffer</i> - This is the pointer to the string of data to be transmitted.
Return Value	None
Remarks:	This function writes the data to be transmitted into the transmit buffer until NULL character is encountered. Once the transmit buffer is full, it waits until data gets transmitted and then writes the next data into the Transmit register.
Source File:	putsUART1.c putsUART2.c
Code Example	putsUART1(Txdata_loc);

3.9 ReadUARTx or getcUARTx

Function Prototype	unsigned int ReadUART1(void); unsigned int ReadUART2(void); unsigned int getcUART1(void); unsigned int getcUART2(void);
Include	uart.h
Description	This function returns the content of UART receive buffer (UxRXREG) register.
Arguments	None
Return Value	This function returns the contents of Receive buffer (UxRXREG) register.
Remarks:	This function returns the contents of the Receive Buffer register. If 9 bit reception is enabled, the entire register content is returned. If 8 bit reception is enabled, then register is read and the 9th bit is masked. Note: getcUARTx () is the same as ReadUARTx ().
Source File:	ReadUART1.c ReadUART2.c
Code Example	unsigned int RX_data; RX_data = ReadUART1();

3.10 WriteUARTx or putcUARTx

Function Prototype	void WriteUART1(unsigned int data); void WriteUART2(unsigned int data); void putcUART1(unsigned int data); void putcUART2(unsigned int data);
Include	uart.h
Description	This function writes data to be transmitted into the transmit buffer

Arguments	(UxTXREG) register.
Return Value	<i>data</i> - This is the data to be transmitted.
Remarks:	None. This function writes the data to be transmitted into the transmit buffer. If 9-bit transmission is enabled, the 9-bit value is written into the transmit buffer. If 8-bit transmission is enabled, then upper byte is masked and then written into the transmit buffer. Note: <code>putcUARTx ()</code> is the same as <code>WriteUARTx ()</code> .
Source File:	<code>WriteUART1.c</code> <code>WriteUART2.c</code>
Code Example	<code>WriteUART1(0xFF);</code>

4 Macros

4.1 *EnableIntUxRX*

Macro	<code>EnableIntU1RX</code> <code>EnableIntU2RX</code>
Overview	Enables UARTx Receive Interrupt.
Input	None
Output	None
Remarks	None
Code Example	<code>EnableIntU1RX;</code>

4.2 *EnableIntUxTX*

Macro	<code>EnableIntU1TX</code> <code>EnableIntU2TX</code>
Overview	Enables UARTx Transmit Interrupt.
Input	None
Output	None
Remarks	None
Code Example	<code>EnableIntU1TX;</code>

4.3 *DisableIntUxRX*

Macro	<code>DisableIntU1RX</code> <code>DisableIntU2RX</code>
Overview	Disables UARTx Receive Interrupt.
Input	None
Output	None
Remarks	None
Code Example	<code>DisableIntU1RX;</code>

4.4 *DisableIntUxTX*

Macro	<code>DisableIntU1TX</code> <code>DisableIntU2TX</code>
Overview	Disables UARTx Transmit Interrupt.
Input	None
Output	None
Remarks	None
Code Example	<code>DisableIntU1TX;</code>

4.5 *SetPriorityIntUxRX*

Macro	SetPriorityIntU1RX SetPriorityIntU2RX
Overview	Sets interrupt priority for UARTx Receive Interrupt.
Input	Interrupt Priority Level.
Output	None
Remarks	None
Code Example	SetPriorityIntU1RX(2);

4.6 SetPriorityIntUxTX

Macro	SetPriorityIntU1TX SetPriorityIntU2TX
Overview	Sets interrupt priority for UARTx Transmit Interrupt.
Input	Interrupt Priority Level.
Output	None
Remarks	None
Code Example	SetPriorityIntU1RX(2);

dsPIC30F/dsPIC33F

Motor Control PWM Peripheral Module Library Help

Table of Contents

- 1 Library Features**
- 2 Using the Library Module in a Project**
- 3 Functions**
 - 3.1 *CloseMCPWM*
 - 3.2 *ConfigIntMCPWM*
 - 3.3 *OpenMCPWM*
 - 3.4 *OverrideMCPWM*
 - 3.5 *SetDCMCPWM*
 - 3.6 *SetMCPWMDeadTimeAssignment*
 - 3.7 *SetMCPWMDeadTimeGeneration*
 - 3.8 *SetMCPWMFaultA*
 - 3.9 *SetMCPWMFaultB*
- 4 Macros**
 - 4.1 *EnableIntMCPWM*
 - 4.2 *DisableIntMCPWM*
 - 4.3 *SetPriorityIntMCPWM*
 - 4.4 *EnableIntFLTA*
 - 4.5 *DisableIntFLTA*
 - 4.6 *SetPriorityIntFLTA*
 - 4.7 *EnableIntFLTB*
 - 4.8 *DisableIntFLTB*
 - 4.9 *SetPriorityIntFLTB*

1 Library Features

This peripheral library module provides:

- Support for all dsPIC30F and dsPIC33F devices in the Motor Control Family.
- Module initialization functions.
- Simple functions to read from and write to the duty cycle and dead time registers.
- Simple interface macros to enable/disable interrupts.

2 Using the Library Module in a Project

Library routine parameters can be constructed using an AND based mask. For more information on this masks, see [16-bit Peripheral Libraries](#). An example of use is shown below.

dsPIC30F Example of Use (AND mask)

```
#define __dsPIC30F6010__
#include <p30fxxxx.h>
#include<pwm.h>
void __attribute__((__interrupt__)) _PWMInterrupt(void)
{
    IFS2bits.PWMIF = 0;
}
int main()
{
    /* Holds the PWM interrupt configuration value*/
    unsigned int config;
    /* Holds the value to be loaded into dutycycle register */
    unsigned int period;
    /* Holds the value to be loaded into special event compare register */
    unsigned int sptime;
    /* Holds PWM configuration value */
    unsigned int config1;
    /* Holds the value be loaded into PWMCON1 register */
    unsigned int config2;
    /* Holds the value to configure the special event trigger
    postscale and dutycycle */
    unsigned int config3;
    /* The value of 'duty cyclereg' determines the duty cycle
    register(PDCx) to be written */
    unsigned int duty cyclereg;
    unsigned int dutycycle;
    unsigned char updatedisable;

    /* Configure pwm interrupt enable/disable and set interrupt
    priorities */
    config = (PWM_INT_EN & PWM_FLTA_DIS_INT & PWM_INT_PR1
              & PWM_FLTA_INT_PRO
              & PWM_FLTB_DIS_INT & PWM_FLTB_INT_PRO);
    ConfigIntMCPWM( config );
    /* Configure PWM to generate square wave of 50% duty cycle */
    duty cyclereg = 1;
    dutycycle = 0x3FFF;
    updatedisable = 0;

    SetDCMCPWM(duty cyclereg,dutycycle,updatedisable);
    period = 0x7fff;
    sptime = 0x0;
    config1 = (PWM_EN & PWM_PTSIDL_DIS & PWM_OP_SCALE16
              & PWM_IPCLK_SCALE16 &
              PWM_MOD_UPDN);
    config2 = (PWM_MOD1_COMP & PWM_PDIS4H & PWM_PDIS3H &
              PWM_PDIS2H & PWM_PEN1H & PWM_PDIS4L &
              PWM_PDIS3L & PWM_PDIS2L & PWM_PEN1L);
```

```

    config3 = (PWM_SEVOPS1 & PWM_OSYNC_PWM & PWM_UEN);
    OpenMCPWM(period, sptime, config1, config2, config3);
    while(1);
}

```

3 Functions

3.1 CloseMCPWM

Function Prototype void CloseMCPWM(void);
Include pwm.h
Description This function turns off the Motor Control PWM module.
Arguments None
Return Value None
Remarks: This function disables the Motor control PWM module and clears the PWM, Fault A and Fault B Interrupt Enable and Flag bits. This function also clears the PTCON, PWMCON1 and PWMCON2 registers.
Source File: CloseMCPWM.c
Code Example CloseMCPWM();

3.2 ConfigIntMCPWM

Function Prototype void ConfigIntMCPWM(unsigned int config);
Include pwm.h
Description This function configures the PWM, Fault A and Fault B Interrupts.
Arguments *config* - PWM interrupt priority and enable/disable information as defined below:
PWM Interrupt Enable/Disable
PWM_INT_EN
PWM_INT_DIS
PWM Interrupt Priority
PWM_INT_PR0
PWM_INT_PR1
PWM_INT_PR2
PWM_INT_PR3
PWM_INT_PR4
PWM_INT_PR5
PWM_INT_PR6
PWM_INT_PR7
Fault A Interrupt Enable/Disable
PWM_FLTA_EN_INT
PWM_FLTA_DIS_INT
Fault A Interrupt Priority
PWM_FLTA_INT_PR0
PWM_FLTA_INT_PR1
PWM_FLTA_INT_PR2
PWM_FLTA_INT_PR3
PWM_FLTA_INT_PR4
PWM_FLTA_INT_PR5
PWM_FLTA_INT_PR6
PWM_FLTA_INT_PR7
Fault B Interrupt Enable/Disable
PWM_FLTB_EN_INT
PWM_FLTB_DIS_INT
Fault B Interrupt Priority
PWM_FLTB_INT_PR0
PWM_FLTB_INT_PR1
PWM_FLTB_INT_PR2
PWM_FLTB_INT_PR3
PWM_FLTB_INT_PR4
PWM_FLTB_INT_PR5

Return Value	None
Remarks:	This function clears the Interrupt Flag bits, sets the interrupt priorities and enables/disables the interrupts.
Source File:	ConfigIntMCPWM.c
Code Example	<pre> ConfigIntMCPWM(PWM_INT_EN & PWM_INT_PR5 & PWM_FLTA_EN_INT & PWM_FLTA_INT_PR6 & PWM_FLTB_EN_INT & PWM_FLTB_INT_PR7); </pre>

3.3 OpenMCPWM

Function Prototype	<pre> void OpenMCPWM (unsigned int period, unsigned int sptime, unsigned int config1, unsigned int config2, unsigned int config3); </pre>
Include	pwm.h
Description	This function configures the Motor Control PWM module.
Arguments	<p><i>period</i> - The PWM timebase period value to be stored in the PTPER SFR.</p> <p><i>sptime</i> - The special event compare value to be stored in SEVTCMP SFR.</p> <p><i>config1</i> - The parameters to be configured in the PTCON register.</p> <p>PWM Module Enable/Disable</p> <pre> PWM_EN PWM_DIS </pre> <p>Idle Mode Operation</p> <pre> PWM_IDLE_STOP PWM_IDLE_CON </pre> <p>Output Postscaler Select</p> <pre> PWM_OP_SCALE1 PWM_OP_SCALE2 PWM_OP_SCALE15 PWM_OP_SCALE16 </pre> <p>Input Prescaler Select</p> <pre> PWM_IPCLK_SCALE1 PWM_IPCLK_SCALE4 PWM_IPCLK_SCALE16 PWM_IPCLK_SCALE64 </pre> <p>PWM Operation Mode</p> <pre> PWM_MOD_FREE PWM_MOD_SING PWM_MOD_UPDN PWM_MOD_DBL </pre> <p><i>config2</i> - The parameters to be configured in the PWMCON1 register as defined below:</p> <p>PWM I/O Pin Pair Complementary/Independent Mode Select</p> <pre> PWM_MOD4_COMP PWM_MOD3_COMP PWM_MOD2_COMP PWM_MOD1_COMP PWM_MOD4_IND PWM_MOD3_IND PWM_MOD2_IND PWM_MOD1_IND </pre> <p>PWM High/Low Output Enable/Disable</p>

```

PWM_PEN4H
PWM_PDIS4H
PWM_PEN3H
PWM_PDIS3H
PWM_PEN2H
PWM_PDIS2H
PWM_PEN1H
PWM_PDIS1H
PWM_PEN4L
PWM_PDIS4L
PWM_PEN3L
PWM_PDIS3L
PWM_PEN2L
PWM_PDIS2L
PWM_PEN1L
PWM_PDIS1L

```

config3 - The parameters to be configured in the PWMCON2 register as defined below:

Special Event Postscaler

```

PWM_SEVOPS1
PWM_SEVOPS2

```

.....

```

PWM_SEVOPS15
PWM_SEVOPS16

```

Output Override Synchronization Mode

```

PWM_OSYNC_PWM
PWM_OSYNC_Tcy

```

PWM Update Enable/Disable

```

PWM_UDIS
PWM_UEN

```

Return Value

None

Remarks:

This function configures the PTPER, SEVTCMP, PTCON, PWMCON1 and PWMCON2 registers. Bit defines related to PWM4 are available only for certain devices and the suitable data sheet should be referred to.

Source File:

OpenMCPWM.c

Code Example

```

period = 0x7fff;
sptime = 0x0;
config1 = PWM_EN & PWM_PTSIDL_DIS & PWM_OP_SCALE16 &
PWM_IPCLK_SCALE16 & PWM_MOD_UPDN;
config2 = PWM_MOD1_COMP & PWM_PDIS4H & PWM_PDIS3H &
PWM_PDIS2H & PWM_PEN1H & PWM_PDIS4L & PWM_PDIS3L &
PWM_PDIS2L & PWM_PEN1L;
config3 = PWM_SEVOPS1 & PWM_OSYNC_PWM & PWM_UEN;
OpenMCPWM (period, sptime, config1, config2, config3);

```

3.4 OverrideMCPWM

Function Prototype

```
void OverrideMCPWM(unsigned int config);
```

Include

pwm.h

Description

This function configures the OVDCON register.

Arguments

config - The parameters to be configured in the OVDCON register as defined below:

Output Controlled by PWM Generator

```

PWM_GEN_4H
PWM_GEN_3H
PWM_GEN_2H
PWM_GEN_1H
PWM_GEN_4L
PWM_GEN_3L
PWM_GEN_2L
PWM_GEN_1L

```

Output Controlled by POUT Bits


```
PWM_POUT_4H
PWM_POUT_4L
PWM_POUT_3H
PWM_POUT_3L
PWM_POUT_2H
PWM_POUT_2L
PWM_POUT_1H
PWM_POUT_1L
```

PWM Manual Output Bits

```
PWM_POUT4H_ACT
PWM_POUT4H_INACT
PWM_POUT4L_ACT
PWM_POUT4L_INACT
PWM_POUT3H_ACT
PWM_POUT3H_INACT
PWM_POUT3L_ACT
PWM_POUT3L_INACT
PWM_POUT2H_ACT
PWM_POUT2H_INACT
PWM_POUT2L_ACT
PWM_POUT2L_INACT
PWM_POUT1H_ACT
PWM_POUT1H_INACT
PWM_POUT1L_ACT
PWM_POUT1L_INACT
```

Return Value

None.

Remarks:

This functions configures the PWM Output Override and Manual Control bits of the OVDCON register. Bit defines related to PWM4 are available only for some devices and the suitable data sheet should be referred to.

Source File:

OverrideMCPWM.c

Code Example

```
config = PWM_GEN_1L & PWM_GEN_1H & PWM_POUT1L_INACT &
PWM_POUT3L_INACT;
OverrideMCPWM(config);
```

3.5 SetDCMCPWM

Function Prototype void SetDCMCPWM (unsigned int *dutycyclereg*,
unsigned int *dutycycle*, char *updatedisable*);

Include pwm.h

Description This function configures the Duty Cycle register and updates the 'PWM Update Disable' bit in the PWMCON2 register.

Arguments *dutycyclereg* - The pointer to the Duty Cycle register.
dutycycle - The value to be stored in the Duty Cycle register.
updatedisable - The value to be loaded into the 'Update Disable' bit of PWMCON2.

Return Value None.

Remarks: None.

Source File: SetDCMCPWM.c

Code Example

```
dutycyclereg = 1;
dutycycle = 0xFFFF;
updatedisable = 0;
SetDCMCPWM (dutycyclereg, dutycycle, updatedisable);
```

3.6 SetMCPWMDeadTimeAssignment

Function Prototype void SetMCPWMDeadTimeAssignment (unsigned int *config*);

Include pwm.h

Description This function configures the assignment of dead-time units to PWM output pairs.

Arguments *config* - The parameters to be configured in the DTCON2 register as defined below:

	Dead-Time Select Bits for PWM1 Signal
	PWM_DTS1A_UA
	PWM_DTS1A_UB
	PWM_DTS1I_UA
	PWM_DTS1I_UB
	Dead-Time Select Bits for PWM2 Signal
	PWM_DTS2A_UA
	PWM_DTS2A_UB
	PWM_DTS2I_UA
	PWM_DTS2I_UB
	Dead-Time Select Bits for PWM3 Signal
	PWM_DTS3A_UA
	PWM_DTS3A_UB
	PWM_DTS3I_UA
	PWM_DTS3I_UB
	Dead-Time Select Bits for PWM4 Signal
	PWM_DTS4A_UA
	PWM_DTS4A_UB
	vPWM_DTS4I_UA
	PWM_DTS4I_UB
Return Value	None.
Remarks:	Bit defines related to PWM4 are available only for certain devices and the suitable data sheet should be referred to.
Source File:	SetMCPWMDeadTimeAssignment.c
Code Example	SetMCPWMDeadTimeAssignment(PWM_DTS3A_UA & PWM_DTS2I_UA & PWM_DTS1I_UA);

3.7 SetMCPWMDeadTimeGeneration

Function Prototype	void SetMCPWMDeadTimeGeneration(unsigned int <i>config</i>);
Include	pwm.h
Description	This function configures dead-time values and clock prescalers.
Arguments	<i>config</i> - The parameters to be configured in the DTCON1 register as defined below:
	Dead-Time Unit A Prescale Select constants
	PWM_DTA0
	PWM_DTA1
	PWM_DTA2

	PWM_DTA62
	PWM_DTA63
	Dead-Time Unit B Prescale Select bits
	PWM_DTBPS8
	PWM_DTBPS4
	PWM_DTBPS2
	PWM_DTBPS1
	Dead-Time Unit A Prescale Select bits
	PWM_DTAPS8
	PWM_DTAPS4
	PWM_DTAPS2
	PWM_DTAPS1
	Dead-Time Unit B Prescale Select constants
	PWM_DTB0
	PWM_DTB1
	PWM_DTB2

	PWM_DTB62
	PWM_DTB63
Return Value	None.
Remarks:	Bit defines related to Dead Time Unit B (DTB) are available only for certain devices and the suitable data sheet should be referred to.
Source File:	SetMCPWMDeadTimeGeneration.c

Code Example `SetMCPWMDeadTimeGeneration (PWM_DTBPS16 & PWM_DT54 & PWM_DTAPS8);`

3.8 *SetMCPWMFaultA*

Function Prototype `void SetMCPWMFaultA(unsigned int config);`
Include `pwm.h`
Description This function configures Fault A Override bits, Fault A Mode bit and Fault Input A Enable bits of PWM.
Arguments *config* - The parameters to be configured in the FLTACON register as defined below:
 Fault Input A PWM Override Value bits
 PWM_OVA4H_ACTIVE
 PWM_OVA3H_ACTIVE
 PWM_OVA2H_ACTIVE
 PWM_OVA1H_ACTIVE
 PWM_OVA4L_ACTIVE
 PWM_OVA3L_ACTIVE
 PWM_OVA2L_ACTIVE
 PWM_OVA1L_ACTIVE
 PWM_OVA4H_INACTIVE
 PWM_OVA3H_INACTIVE
 PWM_OVA2H_INACTIVE
 PWM_OVA1H_INACTIVE
 PWM_OVA4L_INACTIVE
 PWM_OVA3L_INACTIVE
 PWM_OVA2L_INACTIVE
 PWM_OVA1L_INACTIVE
 Fault A Mode bit
 PWM_FLTA_MODE_CYCLE
 PWM_FLTA_MODE_LATCH
 Fault Input A Enable bits
 PWM_FLTA4_EN
 PWM_FLTA4_DIS
 PWM_FLTA3_EN
 PWM_FLTA3_DIS
 PWM_FLTA2_EN
 PWM_FLTA2_DIS
 PWM_FLTA1_EN
 PWM_FLTA1_DIS
Return Value None.
Remarks: Bit defines related to PWM4 is available only for certain devices and the suitable data sheet should be referred to.
Source File: `SetMCPWMFaultA.c`
Code Example `SetMCPWMFaultA (PWM_OVA3L_INACTIVE & PWM_FLTA_MODE_LATCH & PWM_FLTA1_DIS);`

3.9 *SetMCPWMFaultB*

Function Prototype `void SetMCPWMFaultB(unsigned int config);`
Include `pwm.h`
Description This function configures Fault B Override bits, Fault B Mode bit and Fault Input B Enable bits of PWM.
Arguments *config* - The parameters to be configured in the FLTBCON register as defined below:
 Fault Input B PWM Override Value bits
 PWM_OVB4H_ACTIVE
 PWM_OVB3H_ACTIVE
 PWM_OVB2H_ACTIVE
 PWM_OVB1H_ACTIVE
 PWM_OVB4L_ACTIVE

```

PWM_OVB3L_ACTIVE
PWM_OVB2L_ACTIVE
PWM_OVB1L_ACTIVE
PWM_OVB4H_INACTIVE
PWM_OVB3H_INACTIVE
PWM_OVB2H_INACTIVE
PWM_OVB1H_INACTIVE
PWM_OVB4L_INACTIVE
PWM_OVB3L_INACTIVE
PWM_OVB2L_INACTIVE
PWM_OVB1L_INACTIVE

```

Fault B Mode bit

```

PWM_FLTB_MODE_CYCLE
PWM_FLTB_MODE_LATCH

```

Fault Input B Enable bits

```

PWM_FLTB4_EN
PWM_FLTB4_DIS
PWM_FLTB3_EN
PWM_FLTB3_DIS
PWM_FLTB2_EN
PWM_FLTB2_DIS
PWM_FLTB1_EN
PWM_FLTB1_DIS

```

Return Value

None.

Remarks:

FLTBCON register is available only for certain devices and the suitable data sheet should be referred to.

Source File:

SetMCPWMFaultB.c

Code Example

```

SetMCPWMFaultB (PWM_OVB3L_INACTIVE &
PWM_FLTB_MODE_LATCH &
PWM_FLTB2_DIS);

```

4 Macros

4.1 EnableIntMCPWM

Macro	EnableIntMCPWM
Overview	Enables MCPWM Interrupt.
Input	None
Output	None
Remarks	None
Code Example	EnableIntMCPWM;

4.2 DisableIntMCPWM

Macro	DisableIntMCPWM
Overview	Disables MCPWM Interrupt.
Input	None
Output	None
Remarks	None
Code Example	DisableIntMCPWM;

4.3 SetPriorityIntMCPWM

Macro	SetPriorityIntMCPWM
Overview	Sets interrupt priority for MCPWM Interrupt.
Input	Interrupt Priority Level.
Output	None
Remarks	None
Code Example	SetPriorityIntMCPWM(2);

4.4 *EnableIntFLTA*

Macro	<code>EnableIntFLTA</code>
Overview	Enables FLTA Interrupt.
Input	None
Output	None
Remarks	None
Code Example	<code>EnableIntFLTA;</code>

4.5 *DisableIntFLTA*

Macro	<code>DisableIntFLTA</code>
Overview	Disables FLTA Interrupt.
Input	None
Output	None
Remarks	None
Code Example	<code>DisableIntFLTA;</code>

4.6 *SetPriorityIntFLTA*

Macro	<code>SetPriorityIntFLTA</code>
Overview	Sets interrupt priority for FLTA Interrupt.
Input	Interrupt Priority Level.
Output	None
Remarks	None
Code Example	<code>SetPriorityIntFLTA (2);</code>

4.7 *EnableIntFLTB*

Macro	<code>EnableIntFLTB</code>
Overview	Enables FLTB Interrupt.
Input	None
Output	None
Remarks	None
Code Example	<code>EnableIntFLTB;</code>

4.8 *DisableIntFLTB*

Macro	<code>DisableIntFLTB</code>
Overview	Disables FLTB Interrupt.
Input	None
Output	None
Remarks	None
Code Example	<code>DisableIntFLTB;</code>

4.9 *SetPriorityIntFLTB*

Macro	<code>SetPriorityIntFLTB</code>
Overview	Sets interrupt priority for FLTB Interrupt.
Input	Interrupt Priority Level.
Output	None
Remarks	None
Code Example	<code>SetPriorityIntFLTB (2);</code>

dsPIC30F

ADC10 Peripheral Module Library Help

Table of Contents

- 1 Library Features**
- 2 Using the Library Functions in Your Code**
- 3 Functions**
 - 3.1 *BusyADC10*
 - 3.2 *CloseADC10*
 - 3.3 *ConfigIntADC10*
 - 3.4 *ConvertADC10*
 - 3.5 *StopSampADC10*
 - 3.6 *OpenADC10*
 - 3.7 *ReadADC10*
 - 3.8 *SetChanADC10*
- 4 Macros**
 - 4.1 *EnableIntADC*
 - 4.2 *DisableIntADC*
 - 4.3 *SetPriorityIntADC*

1 Library Features

This peripheral library module provides:

- Support for all dsPIC30F Motor Control (MC) devices with ADC modules.
- Module initialization functions.
- Simple functions to start conversions.
- Function to read conversion results and configure interrupts.
- Simple interface macros to enable/disable interrupts.

2 Using the Library Functions in Your Code

Library routine parameters can be constructed using an AND based mask. For more information on this masks, see [16-bit Peripheral Libraries](#). An example of use is shown below.

dsPIC30F Example of Use (AND mask)

```
#define __dsPIC30F6010__
#include <p30fxxx.h>
#include<adc10.h>
unsigned int Channel, PinConfig, Scanselct;
unsigned int Adcon3_reg, Adcon2_reg, Adcon1_reg;

int main(void)
{
    unsigned int result[20], i;
    ADCON1bits.ADON = 0; /* turn off ADC */
    Channel = ADC_CH0_POS_SAMPLEA_AN4 &
              ADC_CH0_NEG_SAMPLEA_NVREF &
              ADC_CH0_POS_SAMPLEB_AN2 &
              ADC_CH0_NEG_SAMPLEB_AN1;
    SetChanADC1(Channel);

    ConfigIntADC10(ADC_INT_DISABLE);
    PinConfig = ENABLE_AN4_ANA;
    Scanselct = SKIP_SCAN_AN2 & SKIP_SCAN_AN5 &
               SKIP_SCAN_AN9 & SKIP_SCAN_AN10 &
               SKIP_SCAN_AN14 & SKIP_SCAN_AN15;

    Adcon3_reg = ADC_SAMPLE_TIME_10 &
                 ADC_CONV_CLK_SYSTEM &
                 ADC_CONV_CLK_13Tcy;

    Adcon2_reg = ADC_VREF_AVDD_AVSS &
                 ADC_SCAN_OFF &
                 ADC_ALT_BUF_OFF &
                 ADC_ALT_INPUT_OFF &
                 ADC_CONVERT_CH0123 &
                 ADC_SAMPLES_PER_INT_16;
    Adcon1_reg = ADC_MODULE_ON &
                 ADC_IDLE_CONTINUE &
                 ADC_FORMAT_INTG &
                 ADC_CLK_MANUAL &
                 ADC_SAMPLE_SIMULTANEOUS &
                 ADC_AUTO_SAMPLING_OFF;
    OpenADC10(Adcon1_reg, Adcon2_reg,
              Adcon3_reg, PinConfig, Scanselct);
    i = 0;
    while(i < 16 )
    {
        ADCON1bits.SAMP = 1;
        while(!ADCON1bits.SAMP);
        ConvertADC10();
    }
}
```

```

        while(ADCON1bits.SAMP);
        while(!BusyADC10());
        while(BusyADC10());
        result[i] = ReadADC10(i);
        i++;
    }
}

```

3 Functions

3.1 *BusyADC10*

Function Prototype	<code>char BusyADC10(void);</code>
Include	<code>adc10.h</code>
Description	This function returns the ADC conversion status.
Arguments	None
Return Value	If the value of DONE is '0', then '1' is returned, indicating that the ADC is busy in conversion. If the value of DONE is '1', then '0' is returned, indicating that the ADC has completed conversion.
Remarks:	This function returns the complement of the ADCON1 <DONE> bit status which indicates whether the ADC is busy in conversion.
Source File:	<code>BusyADC10.c</code>
Code Example	<code>while(BusyADC10());</code>

3.2 *CloseADC10*

Function Prototype	<code>void CloseADC10(void);</code>
Include	<code>adc10.h</code>
Description	This function turns off the ADC module and disables the ADC interrupts.
Arguments	None
Return Value	None
Remarks:	This function first disables the ADC interrupt and then turns off the ADC module. The Interrupt Flag bit (ADIF) is also cleared.
Source File:	<code>CloseADC10.c</code>
Code Example	<code>CloseADC10();</code>

3.3 *ConfigIntADC10*

Function Prototype	<code>void ConfigIntADC10(unsigned int config);</code>
Include	<code>adc10.h</code>
Description	This function configures the ADC interrupt.
Arguments	<i>config</i> - ADC interrupt priority and enable/disable information as defined below: ADC Interrupt enable/disable ADC_INT_ENABLE ADC_INT_DISABLE ADC Interrupt priority ADC_INT_PRI_0 ADC_INT_PRI_1 ADC_INT_PRI_2 ADC_INT_PRI_3 ADC_INT_PRI_4 ADC_INT_PRI_5 ADC_INT_PRI_6 ADC_INT_PRI_7
Return Value	None.
Remarks:	This function clears the Interrupt Flag (ADIF) bit and then sets the interrupt priority and enables/disables the interrupt.

Source File: ConfigIntADC10.c
Code Example ConfigIntADC10(ADC_INT_PRI_3 & ADC_INT_DISABLE);

3.4 ConvertADC10

Function Prototype void ConvertADC10(void);
Include adc10.h
Description This function starts the A/D conversion.
Arguments None
Return Value None
Remarks: This function clears the ADCON1<SAMP> bit and thus stops sampling and starts conversion. This happens only when trigger source for the A/D conversion is selected as Manual, by clearing the ADCON1 <SSRC> bits.
Source File: ConvertADC10.c
Code Example ConvertADC10();

3.5 StopSampADC10

Description This function is identical to ConvertADC10.
Source File: #define to ConvertADC10 in adc10.h.

3.6 OpenADC10

Function Prototype void OpenADC10(unsigned int *config1*,
unsigned int *config2*, unsigned int *config3*,
unsigned int *configport*, unsigned int *configscan*)
Include adc10.h
Description This function configures the ADC.
Arguments *config1* - The parameters to be configured in the ADCON1 register as defined below:
Module On/Off
ADC_MODULE_ON
ADC_MODULE_OFF
Idle mode operation
ADC_IDLE_CONTINUE
ADC_IDLE_STOP
Result output format
ADC_FORMAT_SIGN_FRACT
ADC_FORMAT_FRACT
ADC_FORMAT_SIGN_INT
ADC_FORMAT_INTG
Conversion trigger source
ADC_CLK_AUTO
ADC_CLK_MPWM
ADC_CLK_TMR
ADC_CLK_INT0
ADC_CLK_MANUAL
Auto sampling select
ADC_AUTO_SAMPLING_ON
ADC_AUTO_SAMPLING_OFF
Simultaneous Sampling
ADC_SAMPLE_SIMULTANEOUS
ADC_SAMPLE_INDIVIDUAL
Sample enable
ADC_SAMP_ON
ADC_SAMP_OFF

config2 - The parameters to be configured in the ADCON2 register as defined below:
Voltage Reference

```

ADC_VREF_AVDD_AVSS
ADC_VREF_EXT_AVSS
ADC_VREF_AVDD_EXT
ADC_VREF_EXT_EXT
Scan selection
ADC_SCAN_ON
ADC_SCAN_OFF
A/D channels utilized
ADC_CONVERT_CH0123
ADC_CONVERT_CH01
ADC_CONVERT_CH0
Number of samples between interrupts
ADC_SAMPLES_PER_INT_1
ADC_SAMPLES_PER_INT_2
.....
ADC_SAMPLES_PER_INT_15
ADC_SAMPLES_PER_INT_16
Buffer mode select
ADC_ALT_BUF_ON
ADC_ALT_BUF_OFF
Alternate Input Sample mode select
ADC_ALT_INPUT_ON
ADC_ALT_INPUT_OFF

```

config3 - The parameters to be configured in the ADCON3 register as defined below:

```

Auto Sample Time bits
ADC_SAMPLE_TIME_0
ADC_SAMPLE_TIME_1
.....
ADC_SAMPLE_TIME_30
ADC_SAMPLE_TIME_31
Conversion Clock Source select
ADC_CONV_CLK_INTERNAL_RC
ADC_CONV_CLK_SYSTEM
Conversion clock select
ADC_CONV_CLK_Tcy2
ADC_CONV_CLK_Tcy
ADC_CONV_CLK_3Tcy2
.....
ADC_CONV_CLK_32Tcy

```

configport - The pin select to be configured into the ADPCFG register as defined below:

```

ENABLE_ALL_ANA
ENABLE_ALL_DIG
ENABLE_AN0_ANA
ENABLE_AN1_ANA
ENABLE_AN2_ANA
.....
ENABLE_AN15_ANA

```

configscan - The scan select parameter to be configured into the ADCSSL register as defined below:

```

SCAN_NONE
SCAN_ALL
SKIP_SCAN_AN0
SKIP_SCAN_AN1
.....
SKIP_SCAN_AN15

```

Return Value
Remarks:

None

This function configures the ADC for the following parameters: Operating mode, Sleep mode behavior, Data o/p format, Sample Clk Source, VREF source, No of samples/int, Buffer Fill mode, Alternate I/P sample mod,

	Auto sample time, Conv clock source, Conv Clock Select bits, Port Config Control bits.
Source File:	OpenADC10.c
Code Example	<pre> OpenADC10(ADC_MODULE_OFF & ADC_IDLE_STOP & ADC_FORMAT_SIGN_FRACT & ADC_CLK_INT0 & ADC_SAMPLE_INDIVIDUAL & ADC_AUTO_SAMPLING_ON, ADC_VREF_AVDD_AVSS & ADC_SCAN_OFF & ADC_BUF_MODE_OFF & ADC_ALT_INPUT_ON & ADC_CONVERT_CH0 & ADC_SAMPLES_PER_INT_10, ADC_SAMPLE_TIME_4 & ADC_CONV_CLK_SYSTEM & ADC_CONV_CLK_Tcy, ENABLE_AN1_ANA, SKIP_SCAN_AN0 & SKIP_SCAN_AN3 & SKIP_SCAN_AN4 & SKIP_SCAN_AN5); </pre>

3.7 ReadADC10

Function Prototype	unsigned int ReadADC10(unsigned char <i>bufIndex</i>);
Include	adc10.h
Description	This function reads the ADC Buffer register which contains the conversion value.
Arguments	<i>bufIndex</i> - The ADC buffer number which is to be read.
Return Value	None
Remarks:	This function returns the contents of the ADC Buffer register. You should provide a value between '0' to '15' to ensure a correct read of the ADCBUF0 to ADCBUFF.
Source File:	ReadADC10.c
Code Example	<pre> unsigned int result; result = ReadADC10(3); </pre>

3.8 SetChanADC10

Function Prototype	void SetChanADC10(unsigned int <i>channel</i>);
Include	adc10.h
Description	This function sets the positive and negative inputs for the sample multiplexers A and B.
Arguments	<p><i>channel</i> - The input select parameter to be configured into the ADCHS register as defined below:</p> <p>A/D Channel 1, 2, 3 Negative input for Sample A</p> <pre> ADC_CHX_NEG_SAMPLEA_AN9AN10AN11 ADC_CHX_NEG_SAMPLEA_AN6AN7AN8 ADC_CHX_NEG_SAMPLEA_NVREF </pre> <p>A/D Channel 1, 2, 3 Negative input for Sample B</p> <pre> ADC_CHX_NEG_SAMPLEB_AN9AN10AN11 ADC_CHX_NEG_SAMPLEB_AN6AN7AN8 ADC_CHX_NEG_SAMPLEB_NVREF </pre> <p>A/D Channel 1, 2, 3 Positive input for Sample A</p> <pre> ADC_CHX_POS_SAMPLEA_AN3AN4AN5 ADC_CHX_POS_SAMPLEA_AN0AN1AN2 </pre> <p>A/D Channel 1, 2, 3 Positive input for Sample B</p> <pre> ADC_CHX_POS_SAMPLEA_AN3AN4AN5 ADC_CHX_POS_SAMPLEB_AN0AN1AN2 </pre>

	A/D Channel 0 positive I/P select for Sample A
	ADC_CH0_POS_SAMPLEA_AN0
	ADC_CH0_POS_SAMPLEA_AN1

	ADC_CH0_POS_SAMPLEA_AN15
	A/D Channel 0 negative I/P select for Sample A
	ADC_CH0_NEG_SAMPLEA_AN1
	ADC_CH0_NEG_SAMPLEA_NVREF
	A/D Channel 0 positive I/P select for Sample B
	ADC_CH0_POS_SAMPLEB_AN0
	ADC_CH0_POS_SAMPLEB_AN1

	ADC_CH0_POS_SAMPLEB_AN15
	A/D Channel 0 negative I/P select for Sample B
	ADC_CH0_NEG_SAMPLEB_AN1
	ADC_CH0_NEG_SAMPLEB_NVREF
Return Value	None
Remarks:	This function configures the inputs for sample multiplexers A and B by writing to ADCHS register.
Source File:	SetChanADC10.c
Code Examples	SetChanADC10(ADC_CH0_POS_SAMPLEA_AN0 & ADC_CH0_NEG_SAMPLEA_NVREF) ;

4 Macros

4.1 EnableIntADC

Macro	EnableIntADC
Overview	This macro sets ADC Interrupt Enable bit in Interrupt Enable Control register.
Input	None
Output	None
Remarks	None
Code Example	EnableIntADC;

4.2 DisableIntADC

Macro	DisableIntADC
Overview	This macro clears ADC Interrupt Enable bit in Interrupt Enable Control register.
Input	None
Output	None
Remarks	None
Code Example	DisableIntADC;

4.3 SetPriorityIntADC

Macro	SetPriorityIntADC
Overview	This macro sets ADC Interrupt Priority bits of Interrupt Priority Control register.
Input	Interrupt Priority Level.
Output	None
Remarks	None
Code Example	SetPriorityIntADC(7);

dsPIC30F/dsPIC33F/PIC24H

I/O Ports Peripheral Module Library Help

Table of Contents

- 1 Library Features**
- 2 Using the Library Functions in Your Code**
- 3 Functions**
 - 3.1 *CloseINTx*
 - 3.2 *ConfigINTx*
 - 3.3 *ConfigCNPullups*
 - 3.4 *ConfigIntCN*
- 4 Macros**
 - 4.1 *EnableCNx*
 - 4.2 *DisableCNx*
 - 4.3 *EnableINTx*
 - 4.4 *DisableINTx*
 - 4.5 *SetPriorityIntx*

1 Library Features

This peripheral library module provides:

- Functions to enable and disable I/O port related interrupts.
- Simple interface macros to enable/disable CN and individual external interrupts.

2 Using the Library Functions in Your Code

Library routine parameters can be constructed using an AND based mask. For more information on this masks, see [16-bit Peripheral Libraries](#).

3 Functions

3.1 CloseINTx

Function Prototype	<code>void CloseINT0(void); void CloseINT1(void); void CloseINT2(void); void CloseINT3(void); void CloseINT4(void);</code>
Include	<code>ports.h</code>
Description	This function disables the external interrupt on INT pin.
Arguments	None
Return Value	None
Remarks:	This function disables the interrupt on INT pin and clears the corresponding Interrupt flag.
Source File:	<code>CloseInt0.c CloseInt1.c CloseInt2.c CloseInt3.c CloseInt4.c</code>
Code Example	<code>CloseINT0();</code>

3.2 ConfigINTx

Function Prototype	<code>void ConfigINT0(unsigned int <i>config</i>); void ConfigINT1(unsigned int <i>config</i>); void ConfigINT2(unsigned int <i>config</i>); void ConfigINT3(unsigned int <i>config</i>); void ConfigINT4(unsigned int <i>config</i>);</code>
Include	<code>ports.h</code>
Description	This function configures the interrupt on INT pin.
Arguments	<i>config</i> - Interrupt edge, priority and enable/disable information as defined below: Interrupt edge selection <code>RISING_EDGE_INT</code> <code>FALLING_EDGE_INT</code> Interrupt enable <code>INT_ENABLE</code> <code>INT_DISABLE</code> Interrupt priority <code>INT_PRI_0</code> <code>INT_PRI_1</code> <code>INT_PRI_2</code> <code>INT_PRI_3</code> <code>INT_PRI_4</code> <code>INT_PRI_5</code> <code>INT_PRI_6</code>

	INT_PRI_7
Return Value	None
Remarks:	This function clears the interrupt flag corresponding to the INTx pin and then selects the edge detect polarity. It then sets the interrupt priority and enables/disables the interrupt.
Source File:	ConfigInt0.c ConfigInt1.c ConfigInt2.c ConfigInt3.c ConfigInt4.c
Code Example	ConfigINT0(RISING_EDGE_INT & EXT_INT_PRI_5 & EXT_INT_ENABLE);

3.3 ConfigCNPullups

Function Prototype	void ConfigCNPullups(long int <i>config</i>);
Include	ports.h
Description	This function configures the pull-up resistors for CN pins.
Arguments	<i>config</i> - This is the 32-bit value for configuring pull-ups. The lower word is stored into CNPU1 register and next upper word is stored into CNPU2 register. The upper 8 bits of CNPU2 register are unimplemented.
Return Value	None
Remarks:	None
Source File:	ConfigCNPullups.c
Code Example	ConfigCNPullups(0xFFFF);

3.4 ConfigIntCN

Function Prototype	void ConfigIntCN(long int <i>config</i>);
Include	ports.h
Description	This function configures the CN interrupts.
Arguments	<i>config</i> - This is the 32-bit value for configuring the CN interrupts. The lower 24 bits contain the individual enable/disable information for the CN interrupts. Setting bit x (x = 0, 1, ..., 23) would enable the CNx interrupt. The upper most byte of config contains the Interrupt Priority and Enable/Disable bits. The lower word is stored into the CNEN1 register and next upper byte is stored into the CNEN2 register and the upper most byte is used for setting priority and enable/disable the CN interrupts.
Return Value	None
Remarks:	This function clears the CN interrupt flag and enables/disables the individual interrupts on CN pins. This also configures the interrupt priority and enables/disables the CN Interrupt Enable bit.
Source File:	ConfigIntCN.c
Code Example	// This would enable CN0, CN1, CN2 and CN7 only. ConfigIntCN(CHANGE_INT_OFF & CHANGE_INT_PRI_4 & 0xFF000087);

4 Macros

4.1 EnableCNx

Macro	EnableCN0 EnableCN1 EnableCN2 EnableCN23
Overview	Enables the individual change notification interrupt.

Input	None
Output	None
Remarks	None
Code Example	EnableCN6;

4.2 DisableCNx

Macro	DisableCN0 DisableCN1 DisableCN2 DisableCN23
Overview	Disables the individual change notification interrupt.
Input	None
Output	None
Remarks	None
Code Example	DisableCN14;

4.3 EnableINTx

Macro	EnableINT0 EnableINT1 EnableINT2 EnableINT3 EnableINT4
Overview	Enables the individual external interrupt.
Input	None
Output	None
Remarks	None
Code Example	EnableINT2;

4.4 DisableINTx

Macro	DisableINT0 DisableINT1 DisableINT2 DisableINT3 DisableINT4
Overview	Disables the individual external interrupt.
Input	None
Output	None
Remarks	None
Code Example	DisableINT2;

4.5 SetPriorityIntx

Macro	SetPriorityInt0 SetPriorityInt1 SetPriorityInt2 SetPriorityInt3 SetPriorityInt4
Overview	This macro sets priority for external interrupts.
Input	priority
Output	None
Remarks	This macro sets External Interrupt Priority bits of Interrupt Priority Control register.
Code Example	SetPriorityInt4(6);